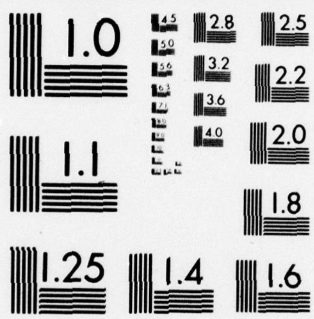
AD-A064 388 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 17/2.1
A BURST ERROR MODEL USING INTERLEAVED BERNOULLI PROCESSES WITH --ETC(U)
DEC 78 R P DAVIS
UNCLASSIFIED AFIT/GE/EE/78-24 NL

1 OF 1
AD
A064388

END
DATE
FILMED
4 -79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

①

LEVEL II

A BURST ERROR MODEL
USING INTERLEAVED BERNOULLI PROCESSES
WITH A MARKOV PROPERTY

THESIS

AFIT/GE/EE/78-24 Robert P. Davis
Capt USAF

14 AFIT/GE/EE/78-24

6 A BURST ERROR MODEL

USING INTERLEAVED BERNOULLI PROCESSES

WITH A MARKOV PROPERTY.

THESIS

9 Master's thesis,

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air Training Command

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

12 79 p.

by

10 Robert P. Davis, ~~B.S.~~

Capt USAF

Graduate Electrical Engineering

11 Dec ~~ember 19~~78

012 225

Preface

I would like to express sincere appreciation to my thesis advisor, Major Joseph Carl, for his many hours of advice, encouragement, and support. I also thank Captain Stan Robinson, Captain Greg Vaughn, and Captain Tom Settecerri for their assistance. A special thanks to my wife, Nancy, for her constant love and support, and for typing this thesis. All of their efforts have been an invaluable help in the development of this thesis.

Contents

List of Figures

List of Tables

Abstract

A model of burst errors from a real-time error bit stream of up to 20 million bits per second is sought. Noise models and channel models are considered. The choice is a two-state Markov channel model in which the states each generated independent Bernoulli random processes. One Bernoulli process produces ones (errors) at a high rate, simulating burst errors. The other process produces ones at a low rate, simulating random errors due to background noise. The transition probabilities determine the average length of the bursts and of the gaps. Relative frequency estimates of the probabilities of certain sequences of one-bits from real data are related to estimates of the model parameters, so relative frequencies provide a basis for fitting this model to real channels using observed error sequences. An equation for the number of errors in a block of bits is developed in terms of the model parameters. Burst probabilities can be predicted based on this equation. The model was tested using computer simulation. Some discussion is devoted to how this burst-error model can be implemented in an actual device to provide real-time channel characterizations. This model aids in the selection of an error correction code.

A BURST ERROR MODEL
USING INTERLEAVED BERNOULLI PROCESSES
WITH A MARKOV PROPERTY

I Introduction

Background

A significant problem in digital communications is the selection of an error correction code. The type of code to use depends on the bit error rate and on the distribution of errors. The distribution is important because errors often occur in groups (burst errors). There are many causes of burst errors in digital communications. Generally, these causes are some form of impulsive noise. On wire lines and telephone networks most impulsive noise is man-made. Some examples are external noise sources such as construction and maintenance work, or internal network noise such as switching (Ref 3:356). Radio communications have some internal noise which can be minimized with design techniques, but the most common cause of impulsive radio noise is lightning (Ref 3:356). The duration of a single lightning strike can affect from a few bits to thousands of consecutive bits, depending on the bit rate of the channel. The intense electrical charge and resulting magnetic fields affect the voltage levels of the received signal, and cause a high error rate in the receiver output. Between lightning strikes, the error rate is very low and is generally caused by Gaussian background noise. If the past several bits are known to be correct or incorrect, the next

bit's probability of being correct could be predicted. This uses the fact that error-free bits come in groups; so if the past several bits are error-free, then the next bit has a high probability of being error-free also. Similarly, if the past bits have a high rate of error, then the next bit has a high probability of being in error. Since the probability of error for the next bit is dependent on the previous bit's correctness, then the error bit stream is said to be dependent.

For uncoded transmissions, the consecutive bits affected by a lightning discharge will not all be processed into error decisions by the receiver. An example would be a receiver using a threshold. The additive noise of the lightning discharge could increase the amplitude of the received signal so that all decisions will be above the threshold. Then, the only errors would be those bits that should have been chosen as below the threshold.

When the burst error characteristics exceed the capability of the error correction code, the decoded data output becomes completely unidentifiable. Thus, the number of bits in a burst error, and the number of error-free bits between bursts, are important distribution parameters. They are needed to implement effective error correction codes (Ref 1:1).

The Problem

The Air Force Communications Service wants a device to monitor an active communication channel and determine its burst error characteristics. Some of these characteristics are the number of bursts per unit time or per interval, the average number of bits per

burst, the maximum number of bits in a burst, the average number of bits between bursts, and minimum number of bits between bursts. The input to this device will be the digital error sequence output of the Hewlett-Packard model HP3761A Error Detector (a stream of bits where a one signifies a bit in error and a zero signifies a correct bit). Two types of output are needed. The first is panel displays of the characteristics that are to be updated at some preset rate determined by the operator. The second type of output would be the characteristics of the entire testing period, usually 24 hours (Ref 1:2).

A capability of testing systems with bit rates up to 20 million bits per second (MBPS) is desired by AFCS. Data reduction is the problem that leads to the choice of these five characteristics by AFCS. However, fully characterizing the burst errors for the entire testing period requires more than the five characteristics discussed above. The most efficient data reduction method would be to represent the stream of error pulses with a mathematical model. This model will completely characterize the error distribution and can be used to determine the higher-order statistics. Defining $P(\underset{\sim}{N} = n)$ for all values of n possible will enable the calculation of the desired statistics. The discrete random variable $\underset{\sim}{N}$ is the number of errors in a block of k bits, and can be any integer, n, from 0 to k. $P(\underset{\sim}{N} = n)$ will be expressed in terms of the model parameters, and thus the higher-order statistics can be determined based on the model.

Such a model will define the device desired by AFCS by determining how to use the HP3761A Error Detector output to calculate $P(\underset{\sim}{N} = n)$. The purpose of this paper will be only the development of the model. The building of an actual device will be left as a follow-on project.

Model Selection

A well designed model will characterize the desired process, enabling prediction of future performance, and ideally, it will model closely the physical nature of the process. The next chapter references the physical nature of burst errors, and discusses an attempt to use a marked point process as a noise model that generates burst errors (Ref 20:128).

$$\underset{\sim}{n}(t) = \sum_{m} \underset{\sim}{W}_m \, \delta(t - \underset{\sim}{T}_m) \qquad (1)$$

Eq (1) is a series of impulses with $\underset{\sim}{W}_m$, a random weight factor, and $\underset{\sim}{T}_m$ the random arrival time of the impulses. This noise model represents additive atmospheric noise where $\underset{\sim}{W}_m$ and $\underset{\sim}{T}_m$ can be selected to produce any pattern of burst error output decisions in a correlation type receiver. The point process model is not mathematically tractable for the application outlined, as will be shown in Chapter II.

The second option abandons the ideal-model concept. Thus, there is only a limited attempt to model the physical nature of the atmospheric noise. The need for simplified math led to the use of a channel model. The channel model characterizes the error bit stream directly, rather than modeling the noise that causes the errors. The selection process, discussed in Chapter III, leads to the two-state first-order Markov model (see Fig 1). The two states are labeled NP1 and NP2. One state, NP2, produces errors at a high rate, simulating burst errors; and the other state, NP1, produces very few errors, simulating the relatively error-free segments between bursts. The transition probabilities from one state to the other are A and B respectively. The terms (1 - A) and (1 - B) are

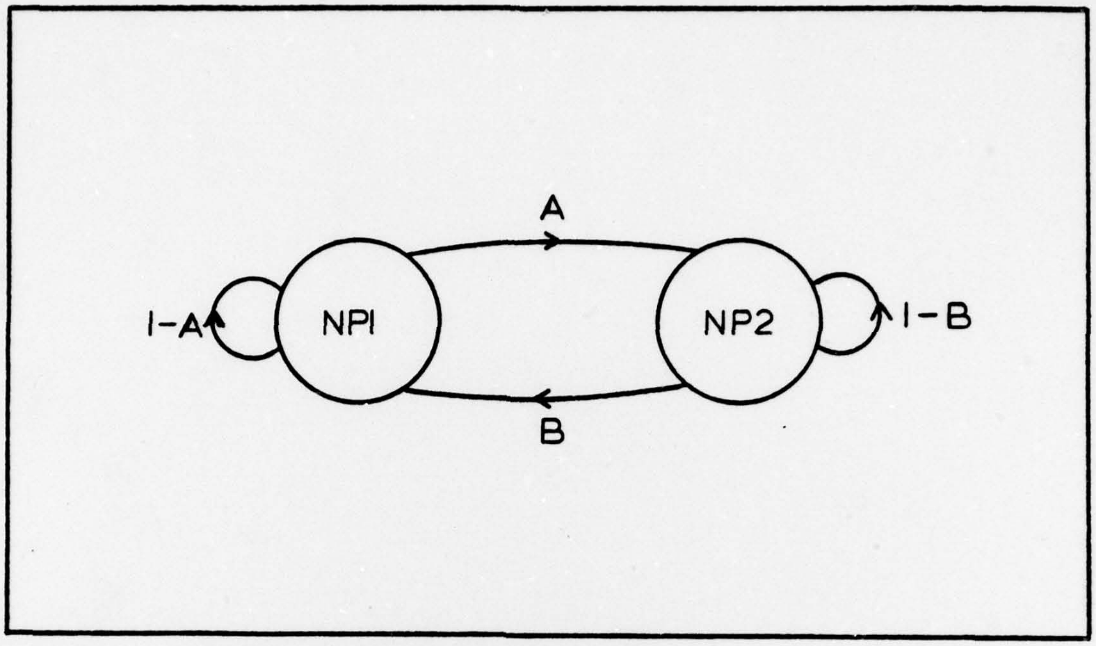


Figure 1. Two-State First-Order Markov Model

the probabilities of staying in a given state from one bit to the next. The error rates of the two states are PE1 for NP1 and PE2 for NP2.

It will be shown that the four model parameters A, B, PE1, and PE2 can be estimated using P(1), P(1,1), P(1,1,1), and P(1,1,1,1), where P(1) is the probability of an error bit, P(1,1) is the joint probability of two successive error bits, P(1,1,1) is the joint probability of three successive error bits, and P(1,1,1,1) is the joint probability of four successive error bits. These four error probabilities are estimated from the error sequence computed by the HP detector during the test. A computer program is used to demonstrate the validity of the estimation procedure and the identification of the model parameters A, B, PE1, and PE2.

Content

The purpose of this paper is to develop a workable model that can be used in the calculation of high-order error statistics during channel tests at high bit rates. Atmospheric noise models are considered first, but the discussion will point out weaknesses that eliminate various noise models. The channel model approach is evaluated, and a two-state first-order Markov model is selected. The main portion of this paper involves details on how the model works, how it fits the proposed application, how to estimate its parameters, evaluating the estimation, and recommendations.

II Modeling Atmospheric Noise

In developing an impulsive noise model, the approach was selected to develop an atmospheric noise model based on lightning. This decision is based on the availability of literature on lightning discharges and the number of models based on lightning. The purpose of this section is to discuss lightning itself to understand what is to be modeled, and then to evaluate the various lightning models. Evaluation will be restricted to comments on strengths and weaknesses, not on proving a model valid or invalid.

Atmospheric Noise

Cloud-to-ground lightning discharges are a primary source of atmospheric radio noise (Ref 14:3). References to lightning in this paper will mean cloud-to-ground discharges. A potential difference between the cloud and ground causes the lightning discharge. The discharge can be broken into two steps: the streamer-leader predischarge and the return discharge. The predischarge is a series of discrete leaders, each covering a short distance (several meters) until a path from the cloud to the ground is formed. Each leader is about one microsecond in duration. The return stroke lasts about 100 microseconds, and goes from the ground back up an ionized path formed by the predischarge (Ref 14:4,5). The field intensity of Fig 2 shows the sequence of leaders followed by the return stroke. The return stroke radiates 95% of the energy of the lightning discharge process. It is the predominate atmospheric noise at very low frequency (VLF) and low frequency (LF), and the leader stroke affects noise in the high frequency (HF) region (Ref 13:3,4).

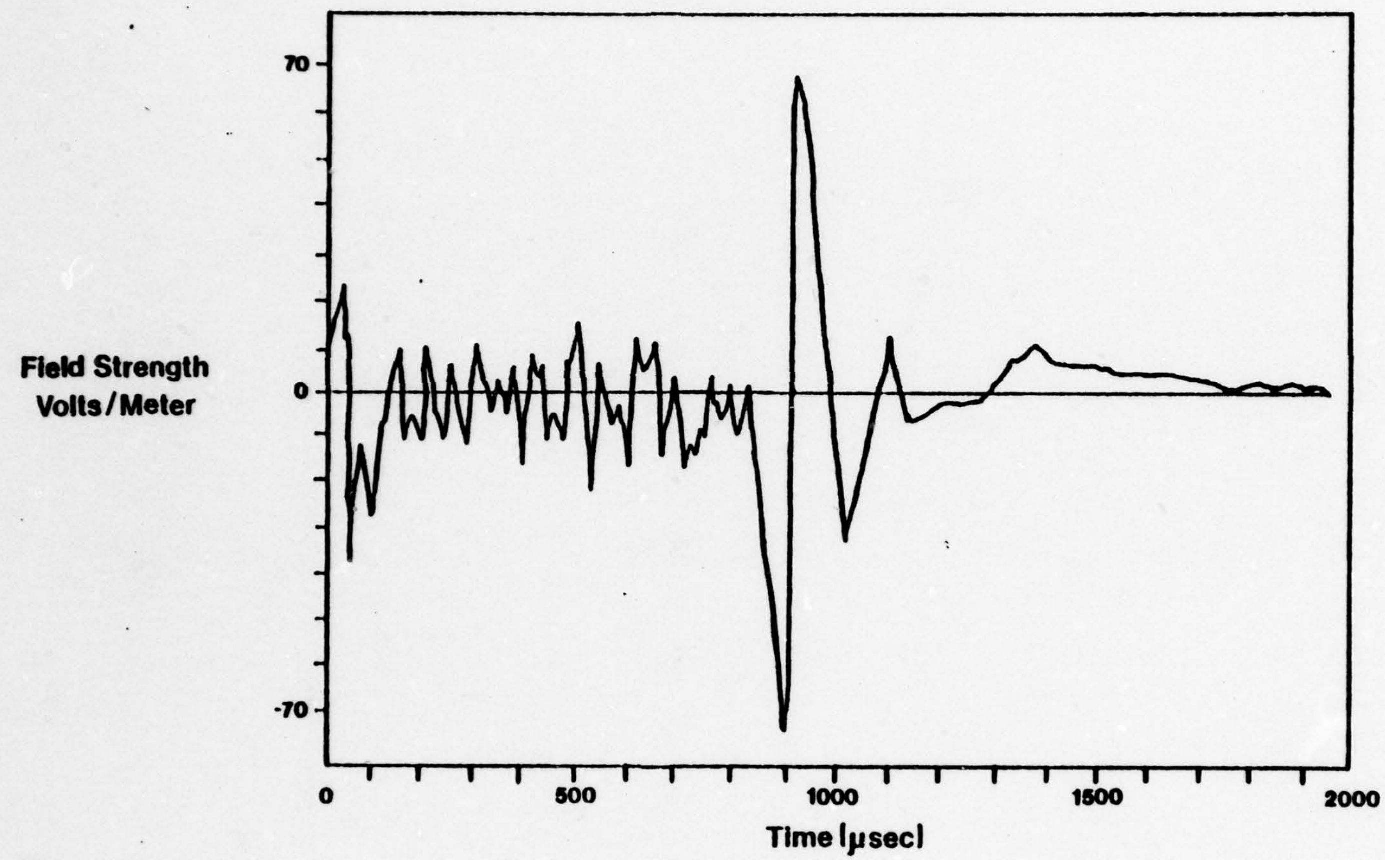


Figure 2. Measured Electric Field from a Lightning Discharge at 20 Km (Ref 13:5)

The 100 microsecond duration of the powerful return stroke can affect a large number of consecutive bits: 1000 bits at a transmission rate of 10 million bits per second. There are many leaders per return stroke, meaning that their effect is shorter, but much more repetitive.

Thousands of storms, creating lightning, exist at any one time all over the earth. Due to its power, duration of discharge, and high occurence rate, lightning is the primary, but not the only, cause of burst errors. The current produced by multiple lightning discharges shown in Fig 3 demonstrates a randomness in the arrival times and the amplitude of the discharges. For the purpose of this paper, atmospheric noise will be considered additive. Several authors give expanded discussions on the physical nature of the lightning discharge (Refs 6;13;14;24;26).

Model Requirements

The intended use of the model places certain restrictions on it. The goal is to produce higher-order statistics to characterize the error distribution. Thus, the first restriction is the ability to calculate higher-order statistics. The second is model simplicity and tractability. This is due to the high bit rates and the need to estimate model parameters from the error bit stream. A complicated model whose parameters can't be estimated is of no value. The third goal is to have the model fit the physical nature of impulsive noise. This last goal is flexible, but is the reason for exploring atmospheric noise models first. Since the atmospheric noise is added to the transmitted signal and then processed by the receiver, simplicity is even more important with noise models. What appears as a tractable

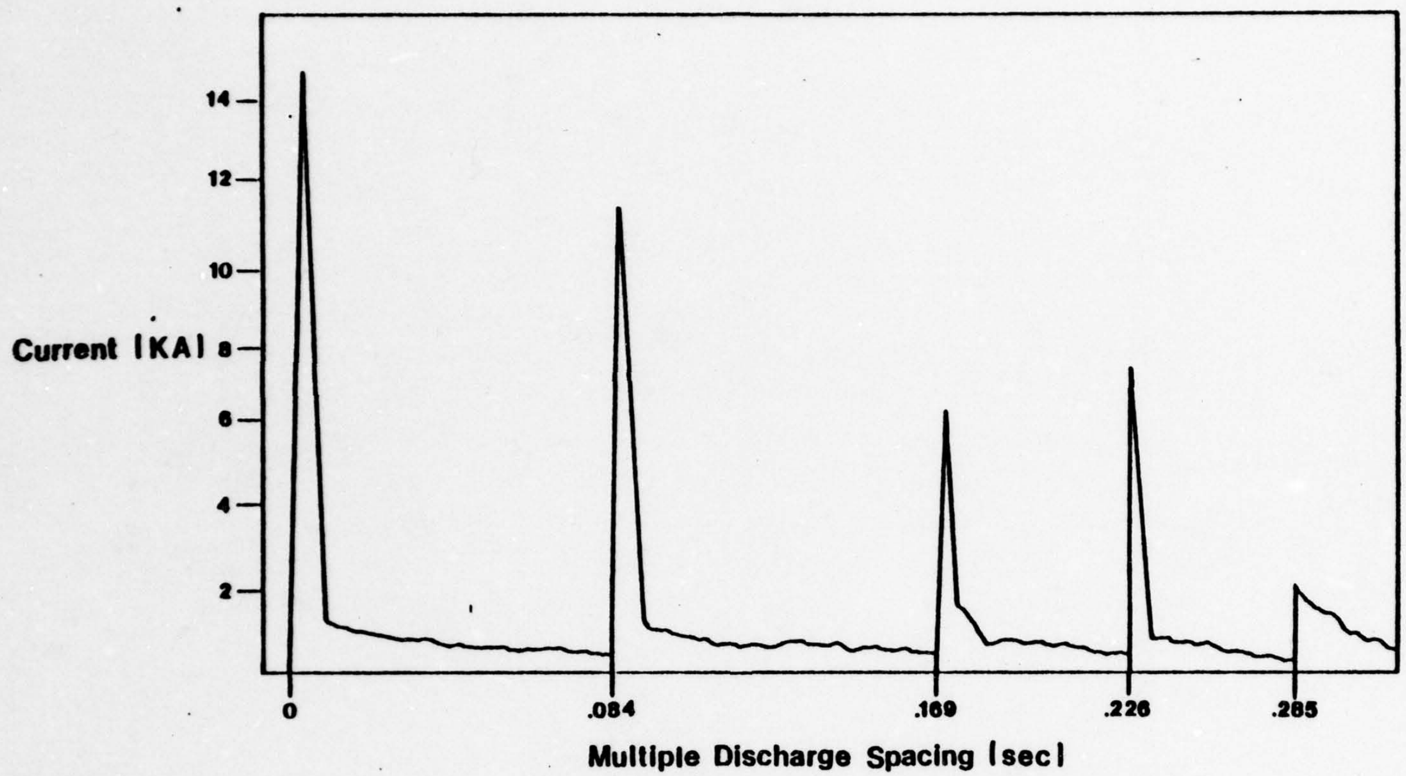


Figure 3. Current Waveform of a Typical Multiple Discharge (Ref 14:7)

model can lose much of its simplicity after the mathematical operations needed to convert it to a receiver output are performed.

Why Use the Point Process Model

There are so many noise models in existence that it is difficult to choose one. The three criteria that have been discussed eliminate many of the existing noise processes. Convenient models, such as Gaussian noise, fit the first two criteria, but must be eliminated because they fail to model the impulsive physical nature of lightning discharges.

Empirical models of lightning discharges (Refs 13;14;15) are designed to fit data compiled on lightning-induced noise. Since these models are based on a tractable matching of first-order statistics, such as APD curves (Refs 2;21), they generally fail at producing higher-order statistics. In addition, most of these curve-fitting models do not model the physical nature of the noise (Refs 13;14).

This leaves the point process, which has the impulses needed, in addition to meeting the first two criteria. The question is, can its impulseness be modified to fit the physical nature of lightning and still meet the first two criteria given above?

The Marked Point Process

The main point of choosing the atmospheric noise approach over channel modeling is to closely model the physical nature of lightning, a major cause of burst errors. The marked point process of Eq (1) is developed by D. L. Snyder (Ref 20). The steps he used to transform a homogenous Poisson process into the marked point process are reviewed here.

The Poisson process can be associated with counting points or impulses, $\sum_{m} \delta(t - \underset{\sim}{T}_m)$. The number of impulses occurring in a time interval (a, b) is Poisson distributed. The number of points in the interval is a random variable, $\underset{\sim}{N}$, with a probability defined as

$$P(\underset{\sim}{N} = n) = \frac{\lambda^n \exp(\lambda)}{n!} \tag{2}$$

where

$$\lambda = \int_a^b \lambda(t)\, dt$$

for all $b \geq a$. The process is homogenous if $\lambda(t)$ is a constant, independent of time. It is an inhomogenous process if $\lambda(t)$ varies with time. λ is the average rate or intensity over the time period (Ref 20:53-54). Snyder defines a compound Poisson process as an inhomogenous Poisson process with independent, identically distributed (iid) marks. This is in the form of Eq (1) with $\underset{\sim}{W}_m$ as the iid marks. The marks are also independent of the counting process arrival times, $\underset{\sim}{T}_m$.

Next, the unmarked Poisson point process is expanded where $\lambda(t)$ depends not only on time, but also on the previous points. This is termed the self-exciting counting process because λ is now a random process that depends on the number of points that have occurred, and can also depend on one or more of the occurence times of the points. The point process is further extended to the doubly stochastic Poisson process where $\lambda(t)$ is a function of $\underline{X}_t$, where $\underline{X}_t$, an external information process, is either a vector-valued Gaussian process or a vector-valued Markovian process. These point processes are in the form of Eq (1) without the marks, $\underset{\sim}{W}_m$.

The marked point process, as defined by Snyder, is a doubly stochastic point process with doubly stochastic marks. This is in the form of Eq (1) (Ref 20).

Selecting the Suitable Point Process

The question is, which of these point processes, if any, meets the three criteria established earlier? To model lightning, as depicted in Fig 1, for bit rates up to 20 MBPS, each return stroke must be broken into a series of amplitude-varying impulses with one in each bit signal duration. This will enable evaluation of the noise effect on each bit. Thus, for a given return stroke, there will be dependence between arrival times and between amplitudes of the points that make up the stroke. This fits the definition of the marked point process. A marked point process model accurately characterizes the physical nature of the noise process, but does not meet the other criteria. The higher-order statistics must be estimated, and Snyder says these calculations can become intractable (Ref 20:460).

Hettinger (Ref 14) suggests using a simpler Poisson point process. He uses two homogenous Poisson point processes, each with iid Gaussian marks. One process has a low rate to simulate lightning pulses, and the other has a high rate to simulate the background noise. His model is tractable, but lacks the dependence between the marks and the dependence of the arrival times on previous points. Thus, its value is limited in the proposed application.

Conclusion

The quick search of atmospheric noise models was desinged to show that the Poisson point process is the best noise model for the

use discussed. However, it has several weaknesses. When the dependence between arrival times and between marks is put into the process, the model appears to be intractable for calculating higher-order statistics. The model proposed by Hettinger is tractable, but in ignoring the dependence in the lightning stroke, the model does not completely model the physical nature of lightning. The physical nature goal is flexible; however, if it is to be relaxed as a criterion, the channel model approach becomes the better choice. The noise model form at the receiver output will be different for each modulation technique and for each receiver type. Thus, the estimation procedure of the model parameters will be different in each case, complicating the problem. The only advantage of the noise model approach was to model the physical nature of lightning. The other criteria have necessitated easing the physical nature goal, so the channel model becomes more practical. It models the error stream directly, and will be in the same form for all noise sources and for all modulation techniques and receiver types.

III Channel Model Selection

There are several terms used in channel model discussions of burst errors. As noted earlier, impulsive noise, such as lightning, does not cause an error in every bit it affects. A definition of what constitutes a burst error is needed. Other terms that must be defined include clusters, gaps, renewal process, and random process.

Several different definitions of burst errors can be found in the literature. Here, a burst is defined as beginning and ending with an error bit. Each error is included in one and only one burst. Each burst is preceded by, and also followed by, a stream of at least L consecutive error-free bits. Thus, bursts are separated by at least L consecutive error-free bits, and no burst can contain a stream of more that (L - 1) consecutive error-free bits. This yields a burst error density, γ, that will be greater than 1/L (Ref 12:1092). Other burst definitions can be found (see Ref 8;11;22;23). This definition was chosen because it allows for error-free bits in the burst. It also appears to be the best definition of this type for implementation in hardware to count bursts and burst lengths.

A cluster is defined to be a run of successive error bits, and does not have any error-free bits. A gap is the run of consecutive error-free bits between error bits. By definition, two consecutive errors have a gap length zero between them (Ref 12:1189). Using this definition, the total number of gaps equals the total number of error bits. Also, gaps can be divided into two groups. The first group consists of the short gaps that exist in bursts. The second group consists of the long gaps that exist in the relatively error-free

bits between bursts. The second group will be called the between burst gaps.

The process definitions relate to the characteristics of the model. A random process is defined as producing independent random errors; thus, it does not produce bursts of dependent error bits. A renewal process is defined as a process where the transition probability to state j of the model, after an error has occurred, is independent of the state in which the error occurred. Therefore, the gaps between errors are independent random variables with the same probability distribution (Ref 17:1713,1715). The errors in a renewal process are dependent. Then a non-renewal process has dependent gaps and dependent errors.

N^{th} Order Markov Models

The dependence between error bits and between error-free bits in the error bit stream has been discussed. N^{th} order means that the probability of error of the next bit depends only on the correctness of the previous N bits. Knowledge of the correctness of bits prior to the previous N bits does not change the error prediction of the next bit. Thus, the probability of an error in the i^{th} bit, given the previous bits, yields

$$P(X_i/X_{i-1}\cdots\cdot X_o) = P(X_i/X_{i-1}\cdots\cdot X_{i-N}) \tag{3}$$

where X_i is the i^{th} bit in the error stream.

Haddad, et al. propose using a Markov gap model (Ref 12). This model assumes that the gap sequence length is a discrete-time, integer-valued Markov process of the first order. The range of

possible gap lengths is divided into sub-ranges so that the sub-ranges are approximately equiprobable. Thus, an infinite dimensional process is reduced to a finite number of states equal to the number of sub-ranges. Manual curve-fitting is required to determine the coefficients of the conditioned gap distributions of the model (Ref 12:1191). This removes the gap model from further consideration.

Another N^{th} order model, the partitioned Markov chain, is proposed by Fritchman (Ref 10). He proposes an N state model partitioned into two types of states. There are M states that produce only error-free bits (gaps) and (N - M) states that produce only error bits (clusters). There are transition probabilities between all states including continuation in the current state. This model has a maximum of 2M(N - M) model parameters (Ref 10:221-225).

Fritchman then simplifies his model to allow it to be specified by the gap distributions. He restricts the model to one error state and (N - 1) error-free states. In additon, he eliminates the transitions between error-free states. The simplified partitioned Markov chain model is depicted in Fig 4 and has a maximum of 2(N - 1) model parameters. Fritchman's models are feasible, if N is small enough to keep the number of parameters reasonable, and will be discussed later.

The Two-State Markov Model

Gilbert (Ref 11) suggested the first model with memory. His two-state Markov model (see Fig 1) has a good state and a bad state. The good state produces only error-free bits, and the bad state yields error bits with a probability $P(e) = 1 - h$. The good state produces

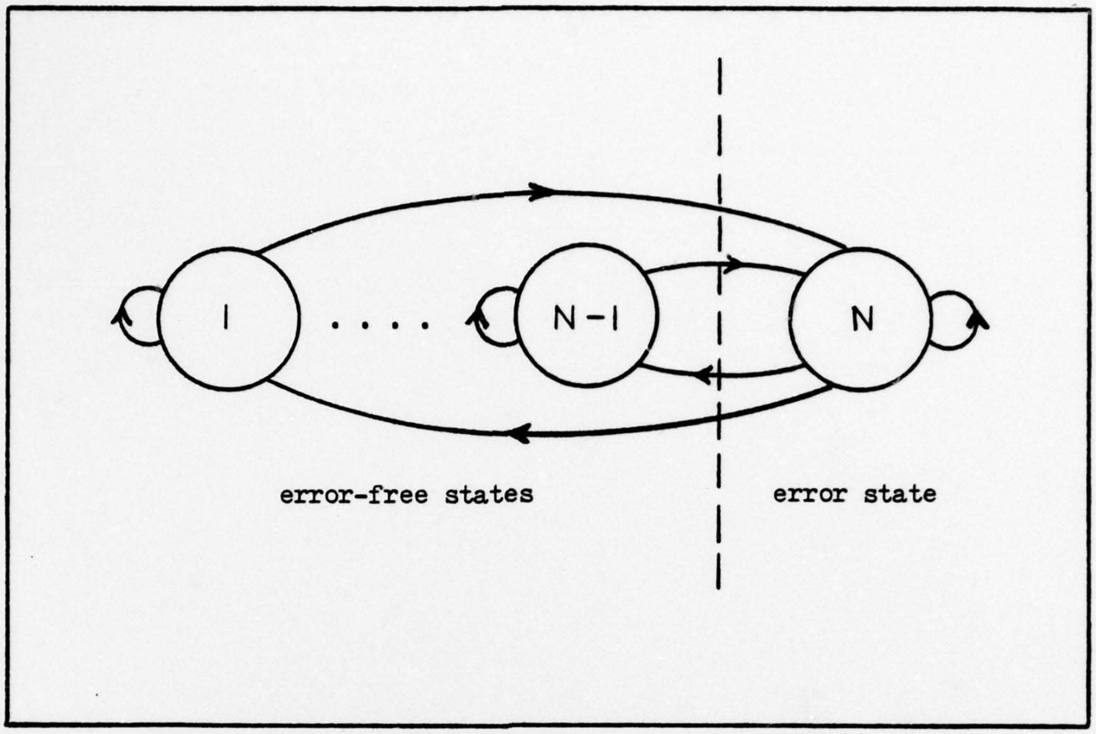


Figure 4. Simplified Partitioned Markov Chain (Ref 10)

the gaps and the bad state produces bursts of density (1 - h). Elliott (Ref 9) generalized Gilbert's model so that each state produces error bits. Using Fig 1 again, the state labeled noise process one (NP1) yields a few random errors at rate PE1 $\ll$ 1. The second state, noise process two (NP2), produces many errors, and can be viewed as a burst of density PE2. The Fritchman models are extensions of these two models. The Gilbert bad state can be broken into two states: one that produces clusters, and one that produces gaps, with h and (1 - h) as the transition probabilities. Then, the Gilbert model looks like the original Fritchman model with two error-free states and one error state. If, in addition, restrictions are included where transitions from the bad state to the good state only occur immediately after an error, and

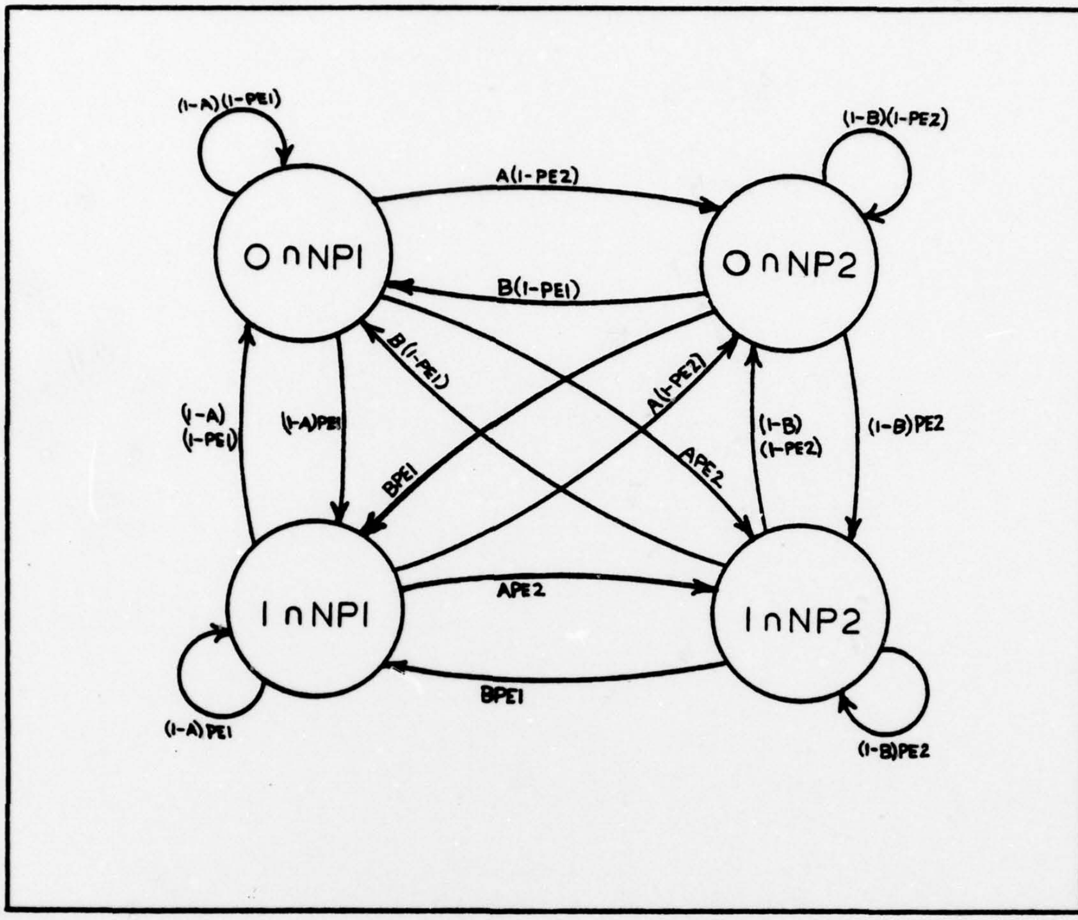


Figure 5. Generalized Gilbert Model in Partitioned Markov Chain form.

transitions from the good state to the bad state only occur immediately before an error, then the Gilbert model becomes a simplified Fritchman model with only one error state. Similarly, if the states of the Elliott model are divided into cluster and gap states, the model looks like the original one proposed by Fritchman with two error states and two error-free states (Fig 5). The states do not generate Bernoulli processes in Fig 5. The top two states only produce zeros and the lower two states only produce ones. Note that the transition probabilities of this model reflect both the transition probabilities and the error

rates of the generalized Gilbert model. The left two states combine to form state NP1 of the generalized Gilbert model, and the right two states combine to form state NP2. There are many models that are variations of Gilbert and generalized Gilbert (Elliott) models (Refs 8;17).

There are several drawbacks to the Gilbert model. Since it has only one error producing state, it is a renewal process; therefore, the gaps are independent. Secondly, it produces only bursts, and does not allow for single, random errors due to background noise. The generalized Gilbert model corrects these flaws. The generalized Gilbert model has two error producing states, and becomes a renewal process only on the conditions that either the transition probabilities $1 - A = B$ and $1 - B = A$ both have equality, or if either error rate is zero. Then, the restrictions that $A + B \neq 1$, $PE1 \neq 0$, and $PE2 \neq 0$ make the gaps dependent random variables with probability distributions that depend on the last gap (Ref 17:1713). Of course, the restrictions $PE \neq PE2$, $A \neq 0$, and $B \neq 0$ must be imposed or the model becomes a random process.

The Generalized Gilbert Model

The generalized Gilbert model of Fig 1 satisfies the model requirements outlined earlier. A detailed discussion of how this model works will illustrate that it meets these requirements. The two states of the model are Bernoulli events. State NP1 produces a sequence of independent bits with $P(x = 1) = PE1$ and $P(x = 0) = 1 - PE1$. State NP2 similarly yields a sequence with $P(x = 1) = PE2$ and $P(x = 0) = 1 - PE2$. The transition probabilities

are stationary, thus the two-state Markov chain is completely specified by the single step transition matrix P

$$P = \begin{bmatrix} 1 - A & A \\ B & 1 - B \end{bmatrix} \tag{4}$$

The key to the generalized Gilbert model is the Bernoulli random variables produced by the two states. This allows the use of a two-state first-order Markov model to simulate an N^{th} order Markov process. Thus, state NP1, which has a very small probability of error, simulates the relatively error-free bits between error bursts, and state NP2, which has a large probability of error, simulates the burst errors. The transition probability $(1 - A)$ of staying in state NP1 determines the approximate number of bits between the bursts. The probability $(1 - B)$ determines the average length of the burst. The states simulate the dependence between error bits, and the transition probabilities determine the Markovian order of the dependence.

The generalized Gilbert model is very flexible. It is a non-renewal model as defined with the restrictions $PE1 \neq PE2$, $A + B \neq 1$, and with none of the parameters zero. However, by allowing $A + B = 1$ or by setting one of the error rates to zero, the model can be converted into a renewal model. The model can also simulate a Bernoulli random process. This can be done by setting $PE1 = 1$ and $PE2 = 0$ and $A + B = 1$. Then this random process has a probability of error $P(e) = B$ and a probability of being correct of $1 - P(e) = A = 1 - B$. These simple conversions mean this model can simulate random, renewal, and non-renewal channels.

The generalized Gilbert model can also be related to the

physical nature of the burst error process. The small probability of error of state NP1 can be translated into errors induced by white Gaussian background noise. The error sequence determined at the receiver output can not differentiate between sources of errors. Thus, errors produced in NP2 can be translated into errors produced by a short duration, intense white Gaussian noise process. Such a simulation of lightning by Gaussian noise is unthinkable in an atmospheric noise model, but here the channel error sequence can not tell the difference. The relationship between probability of error and Gaussian noise is addressed by Van Trees (Ref 25). Thus, the states NP1 and NP2 can be thought of as states of atmospheric noise having the appropriate Gaussian noise parameters associated with them. Being able to view the noise as being created by two Gaussian processes further enhances the simplicity and tractability of the generalized Gilbert model.

Another property of this model is that the two states are first-order Markov. Thus, the probability of any sequence of states occurring can be broken into a product of conditional probabilities by using the Markov property that $P(S_k/S_{k-1},S_{k-2}) = P(S_k/S_{k-1})$. Then, a state sequence $S_kS_{k-1}\cdots S_1$ has probability

$$P(S_kS_{k-1}\cdots S_2S_1) = P(S_k/S_{k-1})P(S_{k-1}/S_{k-2})\cdots P(S_2/S_1)P(S_1) \qquad (5)$$

where S_k is the k^{th} state in the sequence. These conditional state probabilities are the transition probabilities of Fig 1. However, the bit sequence produced by the model is not first-order Markov. The model produces a dependent bit stream where $P(0/0,0) \neq P(0/0)$. This property of the bit stream will be proven in the next chapter.

From the above discussion of Markov channel models it becomes apparent that the generalized Gilbert model is the logical choice. It has great flexibility, and yet is relatively simple. It can simulate non-renewal, renewal, and random channels with only four variable parameters. The next chapter proposes a procedure to estimate the model parameters from the bit stream, and develops an equation for $P(\underset{\sim}{N} = n)$.

IV Estimation of Model Parameters

There are several approaches that can be used as estimation procedures. The one proposed here is relative frequency. The probability of an event can be approximated by the event's relative frequency. The law of large numbers states that the relative frequency of an event can be within any desired accuracy of that event's probability, if the number of trials is made large enough (Ref 18:70-71). The high bit rates and the long testing periods indicate that relative frequency estimation should be well within reasonable limits of accuracy for the stated application. The proposed generalized Gilbert model is defined as stationary, thus the transition probabilities are constant for all time. In addition, the error rates of the two states are independent random variables with fixed probabilities for all time. The model does not change with time, so relative frequency can be used. These are reasonable assumptions for time periods on the order of tens-of-minutes or hours during which millions of samples are available.

Since there are four model parameters, four independent quantities must be estimated from the bit stream. These estimated quantities must be capable of being expressed as equations in terms of the model parameters. The model parameters can then be estimated by solving the four independent equations.

Estimation Procedure

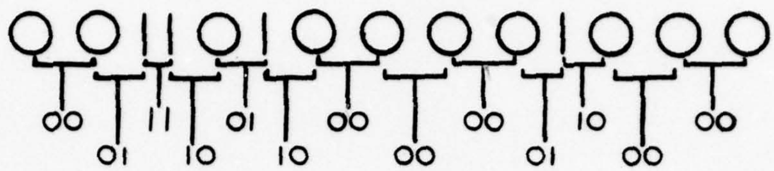
The probability of a correct bit, P(0), and the probability of an error bit, P(1), can be easily estimated from the error sequence using relative frequency. Summing the error-free bits and error bits

respectively and dividing by the total number of bits yields the relative frequency approximation for these probabilities. The joint probability of two successive bits can also be approximated from the error bit sequence. The notation used here for the joint probability of a zero followed by a one in the error stream is P(1,0). This is consistent with the notation used in Eq (3) and Eq (5). The pairwise joint probability estimates are the sum of the overlapping pair combinations in the sequence. For example, 0100··· has 01, 10, 00, 0··· as the overlapping pairs of successive bits. The four possible joint distributions are summed separately and divided by the total number of overlapping pairs (the total number of bits minus one). This yields the relative frequency of the joint probability distributions. Fig 6 illustrates the use of overlapping pair relative frequency. Similarly, the probabilities P(1,1,1) and P(1,1,1,1) could be estimated using overlapping groups of three and four bits respectively.

Probability Equation Development

In developing the equations for these probabilities in terms of the model parameters, several properties used in the model will be needed. First, the two states, NP1 and NP2, are disjoint and therefore, mutually exclusive. Thus, the model can only be in one state at a time, and the transitions between states are considered instantaneous. Second, the model is stationary, and the state transition matrix of Eq (2) defines the model for all time. These two properties mean the probability of beginning in a particular state can be determined from the equations

$$P(NP1 \cup NP2) = P(NP1) + P(NP2) = 1 \tag{6}$$



Total number of bits = 14

Total number of pairs = 13

<u>Relative Frequencies</u>

$\hat{P}(0) = \frac{1}{14} \Sigma$ 0's $= \frac{10}{14}$

$\hat{P}(1) = \frac{1}{14} \Sigma$ 1's $= \frac{4}{14}$

$\hat{P}(0,0) = \frac{1}{13} \Sigma$ 00's $= \frac{6}{13}$

$\hat{P}(1,0) = \frac{1}{13} \Sigma$ 10's $= \frac{3}{13}$

$\hat{P}(0,1) = \frac{1}{13} \Sigma$ 01's $= \frac{3}{13}$

$\hat{P}(1,1) = \frac{1}{13} \Sigma$ 11's $= \frac{1}{13}$

Figure 6. Overlapping Pair Relative Frequency

and

$$P(NP1) = (1 - A)\ P(NP1) + B\ P(NP2) \tag{7}$$

Solving for P(NP1) and P(NP2) yields the steady state probabilities

$$P(NP1) = \frac{B}{A + B} \tag{8}$$

$$P(NP2) = \frac{A}{A + B} \tag{9}$$

These are used as the initial condition probabilities for the starting state.

The notation $0 \cap NP1$ means an error-free bit occurring in state NP1. Similarly, $1 \cap NP2$ means an error bit in state NP2. The joint probability can be expressed as

$$P(0 \cap NP1) = P(0/NP1)\ P(NP1) \tag{10}$$

The occurrence of an error-free bit can be in state NP1 or NP2 and is expressed with the notation $(0 \cap NP1) \cup (0 \cap NP2)$. The probability of an error-free bit is

$$\begin{aligned} P(0) &= P[(0 \cap NP1) \cup (0 \cap NP2)] \\ &= P(0 \cap NP1) + P(0 \cap NP2) \\ &= P(0/NP1)P(NP1) + P(0/NP2)P(NP2) \end{aligned}$$

Substituting the model parameters into this equation using $P(0/NP1) = (1 - PE1)$ and $P(0/NP2) = (1 - PE2)$ yields

$$P(0) = (1 - PE1)\ B/(A + B) + (1 - PE2)A/(A + B) \tag{11}$$

A similar procedure using $P(1/NP1) = PE1$ and $P(1/NP2) = PE2$ leads to

$$P(1) = (PE1)\ B/(A + B) + (PE2)\ A/(A + B) \tag{12}$$

Development of the joint probabilities is more involved, but uses the same properties.

$$\begin{aligned} P(0,0) &= P[(0 \cap NP1) \cup (0 \cap NP2), (0 \cap NP1) \cup (0 \cap NP2)] \\ &= P[(0 \cap NP1) \cup (0 \cap NP2), (0 \cap NP1)] \\ &\qquad + P[(0 \cap NP1) \cup (0 \cap NP2), (0 \cap NP2)] \\ &= P[(0 \cap NP1) \cup (0 \cap NP2)/(0 \cap NP1)]P(0 \cap NP1) \\ &\qquad + P[(0 \cap NP1) \cup (0 \cap NP2)/(0 \cap NP2)]P(0 \cap NP2) \end{aligned}$$

$$= [P(0 \cap NP1/0 \cap NP1) + P(0 \cap NP2/0 \cap NP1)]P(0 \cap NP1)$$

$$+ [P(0 \cap NP1/0 \cap NP2) + P(0 \cap NP2/0 \cap NP2)]P(0 \cap NP2)$$

Since the transition between states is dependent only on the previous state and not on the correctness of the previous bit, then $P(0 \cap NP1/0 \cap NP1) = P(0 \cap NP1/NP1) = P(0/NP1)P(NP1/NP1)$. This yields

$$P(0,0) = [P(0/NP1)P(NP1/NP1) + P(0/NP2)P(NP2/NP1)]P(0/NP1)P(NP1)$$

$$+ [P(0/NP1)P(NP1/NP2) + P(0/NP2)P(NP2/NP2)]P(0/NP2)P(NP2)$$

Substituting the model parameters yields

$$P(0,0) = [(1 - A)(1 - PE1) + A(1 - PE2)](1 - PE1)B/(A + B)$$

$$+ [B(1 - PE1) + (1 - B)(1 - PE2)](1 - PE2)A/(A + B) \quad (13)$$

Similarly, the other joint probabilities are

$$P(0,1) = [(1 - A)(1 - PE1) + A(1 - PE2)](PE1)B/(A + B)$$

$$+ [B(1 - PE1) + (1 - B)(1 - PE2)](PE2)A/(A + B) \quad (14)$$

$$P(1,0) = [(1 - A)PE1 + APE2](1 - PE1)B/(A + B)$$

$$+ [BPE1 + (1 - B)PE2](1 - PE2)A/(A + B) \quad (15)$$

$$P(1,1) = [(1 - A)PE1 + APE2](PE1)B/(A + B)$$

$$+ [BPE1 + (1 - B)PE2](PE2)A/(A + B) \quad (16)$$

These joint probabilities and P(0) and P(1) give six equations, all in terms of the four model parameters. However, they yield only two

independent equations in the four parameters. Since $P(1) = 1 - P(0)$, these two unconditional probabilities yield only one independent equation to solve for the model parameters. The four joint probabilities sum to one, also. In addition, $P(0,1) = P(1,0)$ and $P(0,1) + P(0,0) = P(0)$ and $P(1,0) + P(1,1) = P(1)$; and combining these with either $P(1)$ or $P(0)$ means that only one of the joint probabilities can be used as an independent equation. $P(1)$ and $P(1,1)$ are chosen because they contain PE1 and PE2 rather than 1 - PE1 and 1 - PE2. A set of similar identities show that only one of the three-bit joint probability equations is independent when given $P(1)$ and $P(1,1)$. The identities for the eight three-bit joint probabilities are

$$\Sigma \text{ of all eight probabilities} = 1$$

$$P(1,1,1) + P(1,1,0) = P(1,1)$$

$$P(1,0,1) + P(1,0,0) = P(1,0)$$

$$P(0,1,0) + P(0,1,1) = P(0,0)$$

$$P(0,0,1) + P(0,0,0) = P(0,0)$$

$$P(1,1,0) = P(0,1,1)$$

$$P(0,0,1) = P(1,0,0) \qquad (17)$$

These last two identities and $P(0,1) = P(1,0)$ are as expected. When looking at only two bits, the probability that one of them is an error, and one is not, is the same whether the error occurred first or last. Similarly, when looking at three bits, the probability of two

successive errors and one error-free bit is the same whether the error-free bit occurred before or after the two successive errors. These similar events must not be confused with the dependence that exists in the error stream, $P(1/1) \neq P(1/1,1)$.

The equations for P(1,1,1) and the fourth independent equation, P(1,1,1,1), are developed in the same manner as P(1,1) and yield

$$P(1,1,1) = \{(1 - A)PE1[(1 - A)PE1 + APE2] + APE2[BPE1 + (1 - B)PE2)]\} \cdot BPE1/(A + B)$$
$$+ \{BPE1[(1 - A)PE1 + APE2] + (1 - B)PE2[BPE1 + (1 - B)PE2]\} \cdot APE2/(A + B) \quad (18)$$

and

$$P(1,1,1,1)$$
$$= \{(1 - A)PE1[(1 - A)PE1((1 - A)PE1 + APE2) + APE2(BPE1 + (1 - B)PE2)]$$
$$+ APE2[BPE1((1 - A)PE1 + APE2) + (1 - B)PE2(BPE1 + (1 - B)PE2)]\} \cdot BPE1/(A + B)$$
$$+ \{BPE1[(1 - A)PE1((1 - A)PE1 + APE2) + APE2(BPE1 + (1 - B)PE2)]$$
$$+ (1 - B)PE2[BPE1((1 - A)PE1 + APE2) + (1 - B)PE2(BPE1 + (1 - B)PE2)]\} \cdot APE2/(A + B) \quad (19)$$

Each of the four independent equations has a different order of magnitude in the model parameters. Solving these equations for the

model parameters will yield some high-order polynomials. The problem could be reduced by assuming PE2 to be a constant. For non-renewal channels, PE2 = .5 is the best approximation (Refs 4:627;11:1260;16:60). This assumes that a one and a zero are equiprobable in the message, and the receiver uses some sort of threshold technique. The impulsive noise that causes burst errors would cause the entire received message to be above the receiver threshold during the burst. The receiver output data would be correct approximately half the time (the bits that should have been above threshold), and in error approximately half the time (the bits that should have been below threshold).

Solving Eq (12) for A yields

$$A = \frac{B(PE1 - P(1))}{(P(1) - PE2)} \tag{20}$$

Substituting this into Eq (16) and solving for B yields

$$B = \frac{P(1,1)(PE2 - PE1) + PE1^2(P(1) - PE2) + PE2^2(PE1 - P(1))}{(PE2 - PE1)^2(PE1 - P(1))} \tag{21}$$

Substitution of these values into Eq (17) to solve for PE1 led to a ninth degree polynomial in PE1. Attempts, so far, to reduce this polynomial have failed. It seems reasonable that most of the roots of this polynomial could be eliminated by the restrictions that exist on PE1. First, PE1 must be real and in the range (0,1). In addition, PE2 (assumed to be .5) must be greater than PE1 by several orders of magnitude because the random errors have a much lower density than burst errors. Third, PE1 must be greater than zero or the model becomes a renewal process.

The ninth order equation only arises for non-renewal channels. For random channels PE2 = 0 and PE1 = 1 , so only A and B must be

found. For renewal channels only one error producing state is needed, so PE2 = 0 can be used. PE1 would then simulate the burst errors in the model. Eqs (12) - (17) become simplified. The relative frequency suggested by Gilbert (for his model with only one error state) can be used. With PE2 = 0 the equations for A and B become

$$A = \frac{B(PE1 - P(1))}{P(1)}$$

$$B = \frac{P(1)PE1 - P(1,1)}{PE1 - P(1)}$$

Gilbert uses $C = \frac{P(1,1,1)}{P(1,1,1) + P(1,0,1)}$ to reduce the order of magnitude of the model parameters in his third independent equation. Converting his notation and equation to the form used here yields an equation for PE1 (Ref 11:1260).

$$PE1 = \frac{P(1,1)[2P(1)^2C - P(1,1)[P(1) + C]]}{P(1)^3C - P(1,1)^2} \qquad (22)$$

Thus, the generalized Gilbert model can produce the correct results for renewal processes and random processes as special cases of the parameter values.

An iterative procedure to solve for the model parameters in the general case is discussed in the next chapter.

Two more equations must be developed. First, it should be shown that this model does in fact produce a dependent error bit stream. Second, the goal of this paper is to develop an equation for $P(\underset{\sim}{N} = n)$.

Proof of Dependence

The proof of dependence in the error bit stream requires two inequalities. $P(1) \neq P(1/1)$ shows dependence, but the two-state model for the state sequence is first-order. Therefore, $P(1/1,1) \neq P(1/1)$ is needed to prove that the bit sequence dependency goes beyond first-order. Using Bayes rule, these two inequalities can be converted to $P(1)^2 \neq P(1,1)$ and $P(1,1,1) \neq P(1,1)^2/P(1)$. Multiplying both sides by a constant doesn't change the inequality, but does eliminate the denominator.

$$(A + B)^2 P(1)^2 = (A + B)(BPE1^2 + ABPE1PE2 - ABPE1^2 + APE2^2 + ABPE1PE2 - ABPE2^2)$$

Combining and factoring yields

$$(A + B)^2(P(1,1) - P(1)^2) = (PE2 - PE1)(A)(B)(1 - (A + B)) \neq 0$$

This inequality holds if $PE1 \neq PE2$, $A \neq 0$, $B \neq 0$ and $A + B \neq 1$. These are the same restrictions imposed to prevent the model from becoming a Bernoulli random process. $P(1/1,1) \neq P(1/1)$ becomes

$$(A + B)(APE2 + BPE1)P(1,1,1) \neq (A + B)(APE2 + BPE1)P(1,1)^2/P(1)$$

Multiplying these out, combining and factoring yields

$$(PE1 - PE2)^2[1 - (A + B)]^2 \neq 0$$

This inequality holds if $PE1 \neq PE2$ and $A + B \neq 1$, and both restrictions are already imposed. These inequalities hold for any sequence of bits put into the probability equations. Thus, even

though the state sequence is first-order Markov, the bit sequence has higher-order dependencies - it is not first-order Markov.

Development of $P(\underset{\sim}{N} = n)$

A pattern can be seen in the equations developed for the various joint probabilities, but it isn't a form convenient for notations. Comparing Eqs (11) and (12) to Eqs (13) - (16), and then to Eqs (18) and (19), shows that the addition of a bit adds new terms to the innermost product of the equation. For a given block length, k, $P(\underset{\sim}{N} = n)$ would be the sum of those joint probability equations for k bits that contain n errors. A more convenient notation can be developed.

Using Bayes rule

$$P(\underset{\sim}{N} = n) = \Sigma\, P(\underset{\sim}{N} = n/S_k S_{k-1} \cdots S_1) P(S_k S_{k-1} \cdots S_1)$$

where the summation is over all the 2^k state sequences, $S_k \cdots S_1$. Using the Markov property of the state sequence (see Eq (5)) yields

$$P(\underset{\sim}{N} = n) = \sum_{\substack{2^k \text{ state} \\ \text{sequences}}} P(\underset{\sim}{N} = n/S_k S_{k-1} \cdots S_1) P(S_k/S_{k-1}) \cdots P(S_2/S_1) P(S_1) \quad (23)$$

A convenient notation can be developed for $P(\underset{\sim}{N} = n/S_k \cdots S_1)$. The sum of independent Bernoulli events has a binomial distribution. The two interleaved Bernoulli processes of the model can then be expressed as the product of the two binomial distributions. This is done by dividing the k states into two groups and the n errors into two groups using the following notations.

k = the length of the block of bits

n = the number of errors in the block

k_1 = the number of times state NP1 occurs in a given set of k states

k_2 = the number of times state NP2 occurs in a given set of k states

n_1 = the number of errors that occur in state NP1 in a given set of k states

n_2 = the number of errors that occur in state NP2 in a given set of k states

where

$$k = k_1 + k_2$$

$$n = n_1 + n_2$$

Thus, with this notation, a single summation is used on this product of binomial distributions.

$$P(\underset{\sim}{N} = n/S_k \cdots S_1) = \sum_{n_1 = 0}^{\min(n,k_1)} \binom{k_1}{n_1} PE1^{n_1} (1 - PE1)^{k_1 - n_1} \cdot \binom{k_2}{n_2} PE2^{n_2} (1 - PE2)^{k_2 - n_2} \quad (24)$$

Thus, for a given sequence of states $P(\underset{\sim}{N} = n/S_k \cdots S_1)$ is the average of all the possible ways n errors could occur in the state sequence. The unconditional probability of n errors, $P(\underset{\sim}{N} = n)$, is the conditional probability averaged over all possible state sequences.

This concludes the development of the model and of the estimation procedures. The testing of the model and the relative frequency estimation with computer simulation are discussed in the next chapter.

V Test Results

This chapter discusses the results of three computer programs used to test the model. A computer program has been designed to generate an error stream based on the model parameters. One purpose of this program was to determine if the model would produce a "bursty" error stream. Also, each of the model parameters are varied, one at a time, to show their effect on the bit stream. Second, the program is modified to test the convergence of the relative frequency estimates of P(1), P(1,1), and P(1,1,1). A third program was created to test a simple iterative procedure using successive approximation to estimate the model parameters from the independent equations.

Model Test

The simulation program used to generate an error sequence uses a random number generator function (RANF) that is uniform over the range (0,1). The generated random number is tested against the transition probabilities to determine which state the model is in. If the random number generated is less than or equal to B/(A + B), the initial state is NP1, otherwise it is NP2. Then, a second random number is generated. Based on the state, the random number is tested against the error rate. If the random number is less than the error rate, the bit is a one; if not, it is a zero. After the initial condition, all state transitions are based on the current state. For example, if currently in state NP1, the state will be changed to NP2 if the random number generated is less than or equal to transition probability A; otherwise the state remains NP1. Random numbers are generated to test the state transition and the error rate for every bit in the error stream

simulation. The program, its flow chart, and the error streams it generated can be found in appendix A.

A short test of 2000 bits was used. This was a compromise to have enough bits to detect the burst tendency, and still have a small enough sample to manually analyze the gap distributions. The model parameters were selected to show burst errors and gaps in a small sample, rather than to fit any known channel characteristics. The parameters were $A = .01$, $B = .1$, $PE1 = .001$, and $PE2 = .5$. The values of B and PE2 mean the average burst will be about ten bits long and have a density of about .5. The values of A and PE1 mean the average gap between bursts will be about 100 bits long, and about every tenth between-bursts gap has a random error splitting it into two smaller gaps. The generated error stream would average out to these values if the test sample were large enough (relative frequency). The test data produced very well defined bursts with a density of about .5 and an average length of 9.25 bits. The relatively error-free bits between bursts were well defined also, and averaged 214 bits (due to two gaps of over 400 bits). There were a total of two random errors. The model parameters values tested above will be referred to as the base data test.

To test the model's flexibility, PE1 was increased to .01 and all the other parameters remained the same as in the base data test. The number of random errors increased to eleven. Thus, the error stream was affected as predicted by the change in the model parameter. The average burst length increased to 18 bits, but this was due to one very long burst of 100 bits. The average number of bits between bursts was 125, closer to the expected value with $A = .01$.

In the next test $A = .001$, and the other three parameters were set to the values of the base data test. The average burst length remained the same, the number of random errors went back down to two, but the average number of bits between bursts increased to 400. Again, the bit stream reacted to the parameter change as predicted by the model.

The third test used $B = .01$ with the other three parameters set to the values of the base data test. As expected, the only significant change in the error stream was that the average length of a burst had increased to 110 bits. The fourth test used $PE2 = .9$ as the only change from the base data test. Here, the only significant change was the density of the error burst.

These tests have shown that the model can produce burst errors. In addition, it has shown that each of the model parameters does affect the bit stream as predicted. Thus, by varying the model parameters, any desired distribution of errors can be simulated by the generalized Gilbert model.

The gap distribution is often used to analyze a channel (Refs 9;10;11;12). These tests of the model parameters can be analyzed from that viewpoint. Each test was broken into twenty 100-bit blocks. Thus, the longest possible gap is 100 bits. This was done so that a scale could be used to show the changes in probability for small gap lengths. $P(0^m)$ is the probability of m consecutive error-free bits. $P(m)$ is defined as the probability that the gap length is greater than or equal to m bits. Thus, $P(m)$ is the anticumulative first-order probability distribution (Ref 12:1189-90).

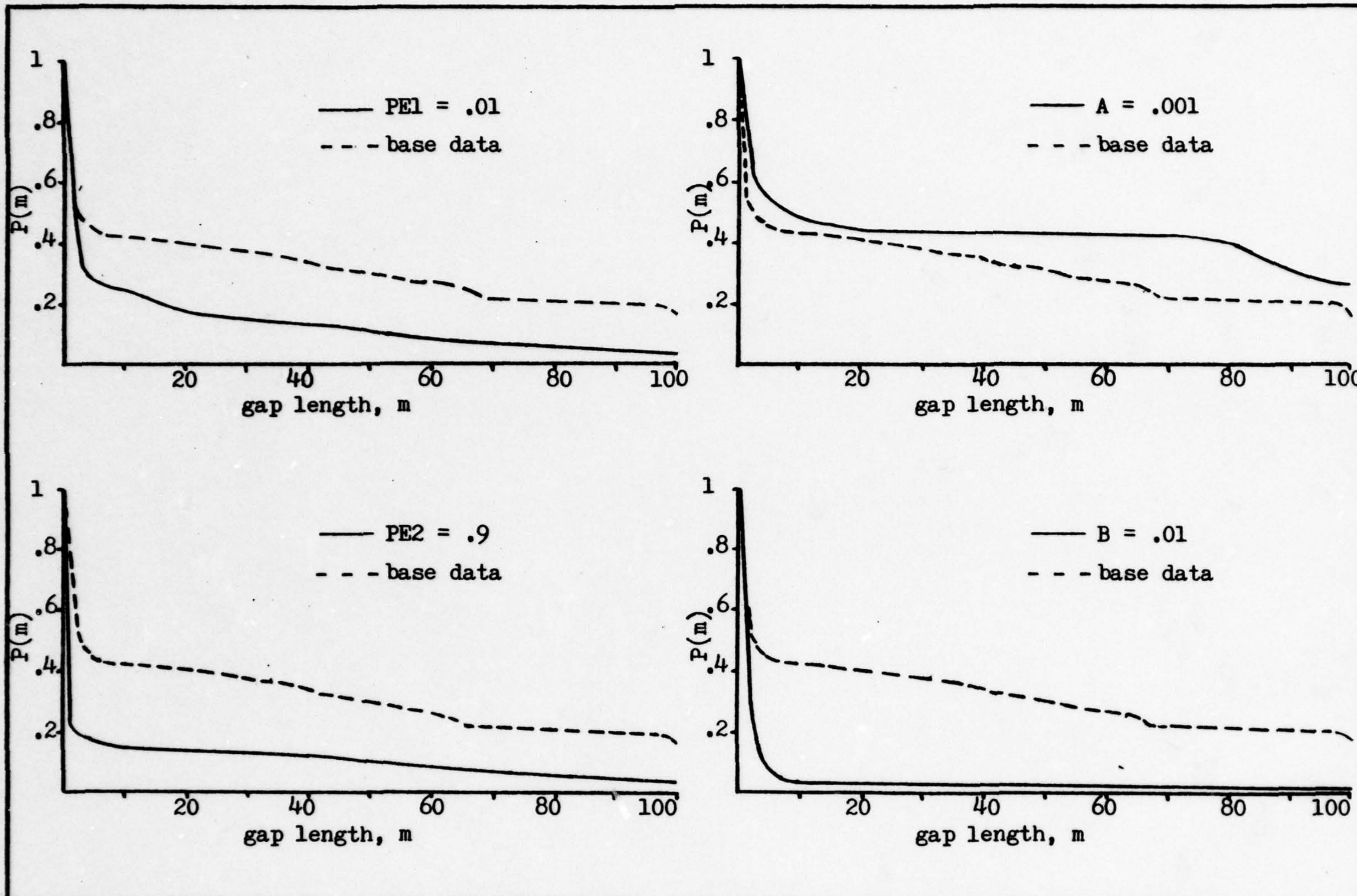


Figure 7. Anticumulative gap distributions, Base Data A = .01, B = .1, PE1 = .001, PE2 = .5

$$P(m) = 1 - \sum_{n=0}^{m-1} P(0^n) \qquad m \geq 1$$

$$P(m) = 1 \qquad m = 0 \qquad (25)$$

Fig 7 is a plot of P(m) versus gap length. Note how the changes in the model parameters change the distribution of the gap lengths. If P(m) were plotted on a log scale, it would have an exponential decay similar to the VHF channel plot in Haddad, et al. (Ref 12:1191). This seems to establish that the model has face validity, but further testing should be done.

Relative Frequency

The same computer simulation was used to generate the error stream based on the model parameters. However, in this program relative frequency was tested for accuracy in estimating P(1), P(1,1) and P(1,1,1). The program and its flow chart can be found in appendix B.

Each bit in the error stream that is a one increments the P(1) counter. The P(1,1) counter is incremented if the current bit and the previous bit are both ones. The P(1,1,1) counter is incremented if the current bit and the two previous bits are ones. These counter values are then divided by the total number of bits, total minus one, and total minus two respectively to get the relative frequency estimates of the probabilities. Table I contains the results of these tests. Note the percent error at 2000 bits is 10 - 19%, but for a one-million bit sample the percent error is reduced to 1 - 2%. Relative frequency should converge much faster than this for a Bernoulli random process. The rather slow convergence indicates the

TABLE I

RELATIVE FREQUENCY TEST

number of bits in sample run	P(1)	P(1,1)	P(1,1,1)
2,000	.0415 (10.49%)	.0165082541 (19.33%)	.0075075075 (18.47%)
50,000	.04768 (2.84%)	.0216604332 (5.84%)	.0099603984 (8.16%)
250,000	.04434 (4.36%)	.0197720791 (3.38%)	.0092360739 (0.3%)
1,000,000	.046894 (1.144%)	.0207420207 (1,356%)	.0093700187 (1.752%)
calculated values	.0463636364 Eq(12)	.0204645364 Eq(16)	.0092086694 Eq(18)

NOTES:

(1) Values in Parentheses are the percent error, using the formula error % = $\frac{C - E1}{C}$ x 100 where C is the calculated value and E is the experimental value.

(2) These tests used a computer-simulated error bit stream based on model parameters A = .01 , B = .1 , PE1 = .001 , and PE2 = .5 .

memory of the model, ie. the dependency in the bit stream. However, relative frequency is a valid estimation procedure for the generalized Gilbert model. Sampling over intervals of several hundred-million bits or more should yield probability estimates with less than 1% error. This seems reasonable for the bit rates and testing periods outlined by AFCS.

The tests conducted to this point indicate that the model performs as predicted, and has the flexibility needed. In addition, the bit stream estimation procedure converges toward the actual probability values, and the sampling intervals should be long enough to insure sufficient accuracy.

The only shortcoming of the model at this point is solving PE1 for non-renewal channels. The next test involves the use of a simple successive approximation routine to solve for PE1.

Successive Approximation

Since PE1 was not readily solvable, this section uses a simple iterative procedure in an attempt to solve the three independent equations for the three unknowns A, B, and PE1 (PE2 = .5). However, since PE1 is a polynomial, there may be several values of PE1 that will solve the equations. Thus, there is some concern that the solution attained here will not be unique.

An equation in the form $f(x) = 0$ can be rewritten as $x = F(x)$. To solve for the roots of $f(x)$ put an initial approximation, x_1, into $F(x)$ and solve for x_2, the second approximation. x_2 is then put into $F(x)$ which yields the third approximation, x_3.

$$x_{j+1} = F(x_j) \tag{26}$$

This successive approximation will converge if $|F'(x)| < 1$ (Ref 5:168).

The equations for P(1), P(1,1), and P(1,1,1) can be put in the form $f_1(A,B,PE1) = 0$, $f_2(A,B,PE1) = 0$ and $f_3(A,B,PE1) = 0$ and then rewritten as

$$A = F_1(A,B,PE1)$$

$$B = F_2(A,B,PE1)$$

$$PE1 = F_3(A,B,PE1)$$

For the non-renewal case, these three model parameters must be less than one and greater than zero. Thus, bounds can be placed on the approximations to aid convergence. These procedures were implemented with a computer program, but it failed to converge.

Eq (26) can be rewritten as

$$x_{j+1} = (1 - H)x_j + HF(x_j) \tag{27}$$

where the convergence could be enhanced by the choice of H (Ref 5:168). In addition, tighter bounds can be placed on the model parameters. It is reasonable to assume that PE1, the random error rate, will be at least two orders of magnitude less than PE2, the burst error rate or density. Also, A should be at least one order of magnitude less than B, meaning the gaps between bursts will be at least ten times longer than the bursts. These changes are incorporated in the computer program in appendix C. AHAT, BHAT, and E1HAT are the symbols used for $F_1(A,B,PE1)$, $F_2(A,B,PE1)$, and $F_3(A,B,PE1)$ respectively. The program is a simple loop computing new approximations for the parameters, and substituting them back into the equations. This program uses the

calculated values for P(1), P(1,1), and P(1,1,1) in the iteration equations. These values are the same ones used in the base data test and the relative frequency test, except $A = .007$.

Fifty iterations were used to determine if the approximations were converging or diverging. Various initial approximations within the parameter bounds were tested. Various values of H were also tested. It was found that $H < 10^{-4}$ kept the parameter approximations within the bounds. However, only two of the three parameters would converge at the same time to the known model parameter values. It appears that the existence of more than one solution to PE1 is responsible for the failure of this iteration procedure. There are a multitude of iteration procedures that could be attempted. However, there is insufficient time to search for one that will work.

The model testing and relative frequency testing indicate that the generalized Gilbert model can be used for real-time burst error modeling. The iteration procedure did not succeed, but there are other iterative routines that should be attempted. They will be discussed in the conclusions and recommendations that follow.

VI Conclusions and Recommendations

The discussions in this paper lead to several conclusions concerning real-time burst error modeling. Based on these conclusions, some recommendations are made for continued study on this topic. Conclusions are discussed first, covering what has been accomplished. Recommendations are in the last section. They suggest possible approaches for further work.

Conclusions

One of the main thrusts of this paper was to find a suitable model for burst error analysis on a real-time basis. The intended use of the model created restrictions on the model choice. The best choice, based on the criteria, is the generalized Gilbert model. This is a rather uncomplicated model that produces the dependency that exists in the bursts and the dependency that exists in the gaps. Yet, the model is flexible and can model renewal channels and random channels as special cases of the general model.

There is another advantage of this model. Because it doesn't vary with time, relative frequency can be used to estimate probabilities from the real-time bit stream. The probabilities chosen can be easily obtained from bit streams at bit rates up to 20 MBPS. The simple logic circuits needed to increment the counters are shown in Fig 8. Since reliable estimates can be formed in a time period on the order of a minute or less, the stationarity assumption seems justifiable.

The next step was to define these probabilities in terms of the model parameters. This was accomplished, and it was determined that PE2 must be fixed to retain tractability. For many non-renewal

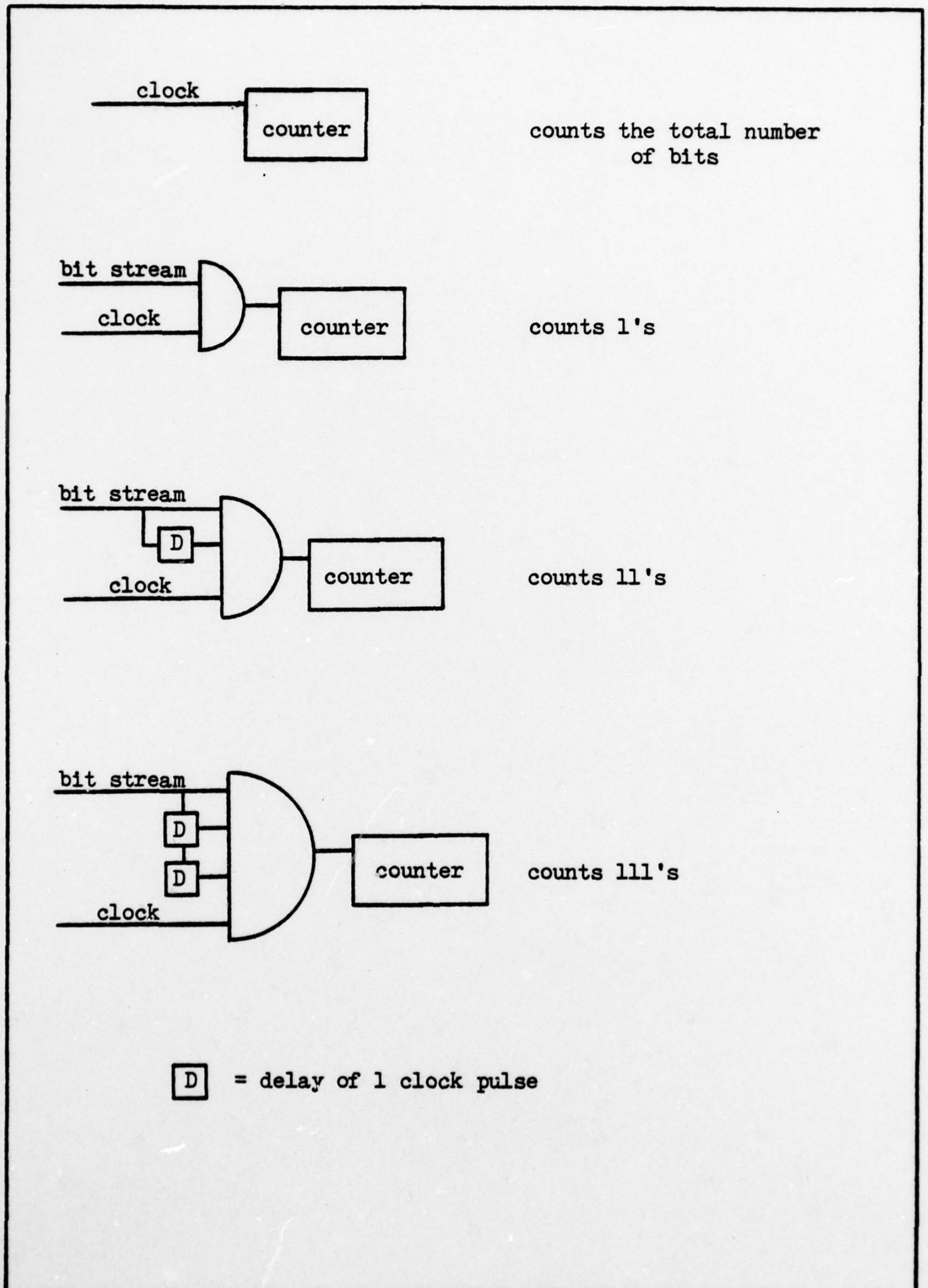


Figure 8. Example Circuits for Relative Frequency Counters.

channels, it appears that PE2 = .5 is the logical choice. For renewal and random channels PE2 = 0 can be used. The one flaw at this point was solving for the three model parameters from the probability equations for non-renewal channels. No suitable method of reducing the polynomial in PE1 was found. The iterative solution attempted in Chapter V failed also.

Recent developments indicate two iteration procedures that may solve the problem of estimating the model parameters A, B, and PE1 from the three independent equations. One approach would be to use the Newton-Raphson iteration. This requires solving the inverse Jacobian of the three equations to be used in the iteration equation (Ref 5:319). The second approach uses a subroutine in the AFIT library. The subroutine MINUM is part of the BINDECK library of subroutines, and is based on a Sperry Rand Report (Ref 19). It uses a combination of random steps, gradient steps, averaging steps, and jump steps. This procedure was tested once using the same model parameter values of the previous iteration technique in Chapter V. The results converged to within 0.1% of the actual parameter values using 2000 iterations.

The other goal was to define $P(\underset{\sim}{N} = n)$ in terms of the model parameters. This probability equation allows calculation of all the burst statistics. $P(\underset{\sim}{N} = n)$ is an important statistic in selecting an error correcting code (Refs 4;9). The equation developed for $P(\underset{\sim}{N} = n)$ in Eq (23) and Eq (24) appears formidable for large block lengths, k. The summation over 2^k state sequences doubles for each bit added to the block length. But this is misleading. Many of these 2^k state sequences are equivalent. The joint probabilities

developed earlier are a good example. The probability of one error in a two bit block is $P(0,1) + P(1,0)$. However, as shown earlier, $P(0,1) = P(1,0)$. Thus, only one event need be calculated. Similarly, $P(1,0,0) = P(0,0,1)$, so the probability of one error in a three bit block would require the calculation of only two separate events, $2P(1,0,0)$ and $P(0,1,0)$. Thus, an algorithm could be developed to take advantage of this property and reduce the calculations considerably. Time is a factor in the relative frequency estimates, but not much of a factor here. $P(\underset{\sim}{N} = n)$ would only be computed periodically during the test period, allowing at least five or ten minutes for its calculation. Specialized hardware could further speed up the process. In fact, the relative frequency estimates compiled during each five to ten minute segment could be stored in memory. Thus, there need not be any time constraint on calculating $P(\underset{\sim}{N} = n)$ for each five or ten minute update period.

The final conclusion is that the generalized Gilbert model seems capable of modeling all types of channels on a real-time basis. It has the potential of becoming a valuable asset in channel analysis.

Recommendations

There are several areas that merit further study. First, additional investigation into reducing the polynomial in PE1 should be made. Once this is done, the model should be tested with a real data stream. The development of an algorithm to calculate $P(\underset{\sim}{N} = n)$ would be the next step, followed by the building of an actual device.

The relationship used by Gilbert to reduce the polynomial in PE1 for his renewal model (PE2 = 0) leads to the belief that some

such relationship exists that would reduce the polynomial in PE1 for the non-renewal model. Even the reduction of this polynomial to a cubic or quadratic would make the problem workable.

The testing done on the model in this paper was all computer simulation. Thus, the second step of the continuation should be to determine the model parameters from the error stream of an actual communications channel. The model parameters generated by the test could then be used in the simulation program. The relative frequency estimates from the actual channel could be compared to those of the simulation. They would determine if the estimated model parameters generate an error stream similar to the actual error stream.

The third step would be to develop an algorithm for computing $P(\underset{\sim}{N} = n)$ for any desired block length, k. The algorithm would be based on Eqs (23) and (24), and could incorporate some of the properties of the joint probabilities that were discussed earlier. This algorithm should be designed for use in an existing micro-processor, or perhaps for compatability with some special purpose hardware.

The iteration procedures discussed in the conclusions as potential solutions should be investigated further. It appears that they could be implemented easily in the microprocessor needed to compute $P(\underset{\sim}{N} = n)$. An iteration procedure should be used if the PE1 polynomial can't be reduced to a lower-order.

The last step in completing the research would be the construction of an actual device. This device would use the error bit stream as an input. The bit stream would use logic circuits of the type shown in Fig 8 to estimate the probabilities P(1), P(1,1), and P(1,1,1)

by relative frequency. From these estimates, A, B, and PE1 will be estimated based on the equations developed here, and the reduced polynomial of PE1 recommended in step one. The algorithm of step three will then calculate the $P(\underset{\sim}{N} = n)$ for a specified block length (determined by an external switch or dial). This device should not be overly expensive because of its simplicity, and could be of great value to AFCS during their channel tests.

Bibliography

1. AFCS/EETDD. Technical Report on the Measurement of Burst Errors in Digital Communications Systems. Defines requirement for error measurement device. Scott AFB, Missouri: Air Force Communications Service, March 1977.

2. Beckman, Petr. "Amplitude-Probability Distribution of Atmospheric Radio Noise," Radio Science Journal of Research, 68D (6): 723-736 (June 1964).

3. Bodonyi, A. B. "Effects of Impulse Noise on Digital Data Transmission," IRE Trans. on Communications Systems, CS-9: 355-361 (December 1961).

4. Cain, J. B., and R. S. Simpson. "The Distribution of Burst Lengths on a Gilbert Channel," IEEE Trans. on Information Theory, (Corres.), IT-15-5: 624-627 (September 1969).

5. Carnahan, Brice, et al. Applied Numerical Methods. New York: John Wiley and Sons, Inc., 1969.

6. Crichlow, W. Q., et al. Amplitude Probability Distributions for Atmospheric Radio Noise. National Bureau of Standards Monograph 23, PB183992. Washington, D. C.: Department of Commerce, November 1960.

7. Davenport, Wilbur B., Jr. Probability and Random Processes. New York: McGraw-Hill, Inc., 1970.

8. Deily, G. R. "Some Properties of the Two-State First Order Markov Chain," IEEE Trans. on Communication Technology, COM-14: 526-529 (August 1966).

9. Elliott, E. O. "Estimates of Error Rates for Codes on Burst Error Channels," Bell Sys. Tech. J., 42: 1977-1997 (September 1963).

10. Fritchman, Bruce D. "A Binary Channel Characterization Using Partitioned Markov Chains," IEEE Trans. on Information Theory, IT-13-2: 221-227 (April 1967).

11. Gilbert, E. N. "Capacity of a Burst-Noise Channel," Bell Sys. Tech. J., 39: 1253-1265 (September 1960).

12. Haddad, A. H., et al. "Markov Gap Models for Real Communication Channels," IEEE Trans. on Communications, COM-23: 1189-1197 (November 1975).

13. Hall, H. M. A New Model for "Impulsive Phenomena": Applications to Atmospheric-Noise Communication Channels. SU-SEL-66-052. Stanford, California: Systems Theory Laboratory, April, 1966. (AD 648650).

14. Hettinger, Stephen D. A Random Point Process Model for Atmospheric Radio Noise. MS Thesis: Wright-Patterson AFB, Ohio: Air Force Institute of Technology, March 1978.

15. Ibukun, O. "Structural Aspects of Atmospheric Radio Noise in the Tropics," Proc. of IEEE, 54: 361-367 (March 1966).

16. Large, Robert W. On the Performance-Index Modeling of Digital Communications Channels. MS Thesis: Wright-Patterson AFB, Ohio: Air Force Institute of Technology, March 1970. (AD 874 441).

17. McCullough, Richard H. "The Binary Regenerative Channel," Bell Sys. Tech. J., 47: 1713-1735 (October 1968).

18. Papoulis, Anthanasios. Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill Book Company, 1965.

19. Pearson, C. E. and W. S. Willman. A Computer Program for Design Optimization. Report No. SRRC-RR-65-48. Sudbury, Massachusetts: Sperry Rand Research Center, June 1965.

20. Snyder, Donald L. Random Point Processes. New York: John Wiley and Sons, Inc., 1975.

21. Spaulding, A. D. "Conversion of the Amplitude-Probability Distribution of Atmospheric Radio Noise from One Bandwidth to Another," Journal of Research of National Bureau of Standards--D. Radio Propagation, 66D (6): 713-720 (1962).

22. Tsai, S. "Evaluation of Burst Error Correcting Codes Using a Simple Partitioned Markov Chain Model," IEEE Trans. on Communications, COM-21: 1031-1034 (September 1973).

23. -----. "Markov Characterization of the HF Channel," IEEE Trans. on Communication Technology, COM-17: 24-32 (February 1969).

24. Uman, M. A. Lightning. New York: McGraw-Hill Book Company, 1969.

25. Van Trees, Harry L. Detection, Estimation and Modulation Theory, Part I. New York: John Wiley and Sons Inc., 1968.

26. Watt, A. D. and E. L. Maxwell. "Characteristics of Atmospheric Noise from 1 to 100 kc," Proc. of IRE, 45: 787-794 (June 1957).

Appendix A

Program to Simulate an Error Bit Stream

The program contained in this appendix uses given values for the model parameters to generate a bit stream based on the model. A simplified flow chart is presented also. Figures (9) through (13) are the actual bit streams generated. The symbols used in the program are defined.

GA = model parameter A, obtained from the data card

GB = model parameter B, obtained from the data card

PE1 = model parameter PE1, obtained from the data card

PE2 = model parameter PE2, obtained from the data card

RVN = randomly generated number to determine the state of the model

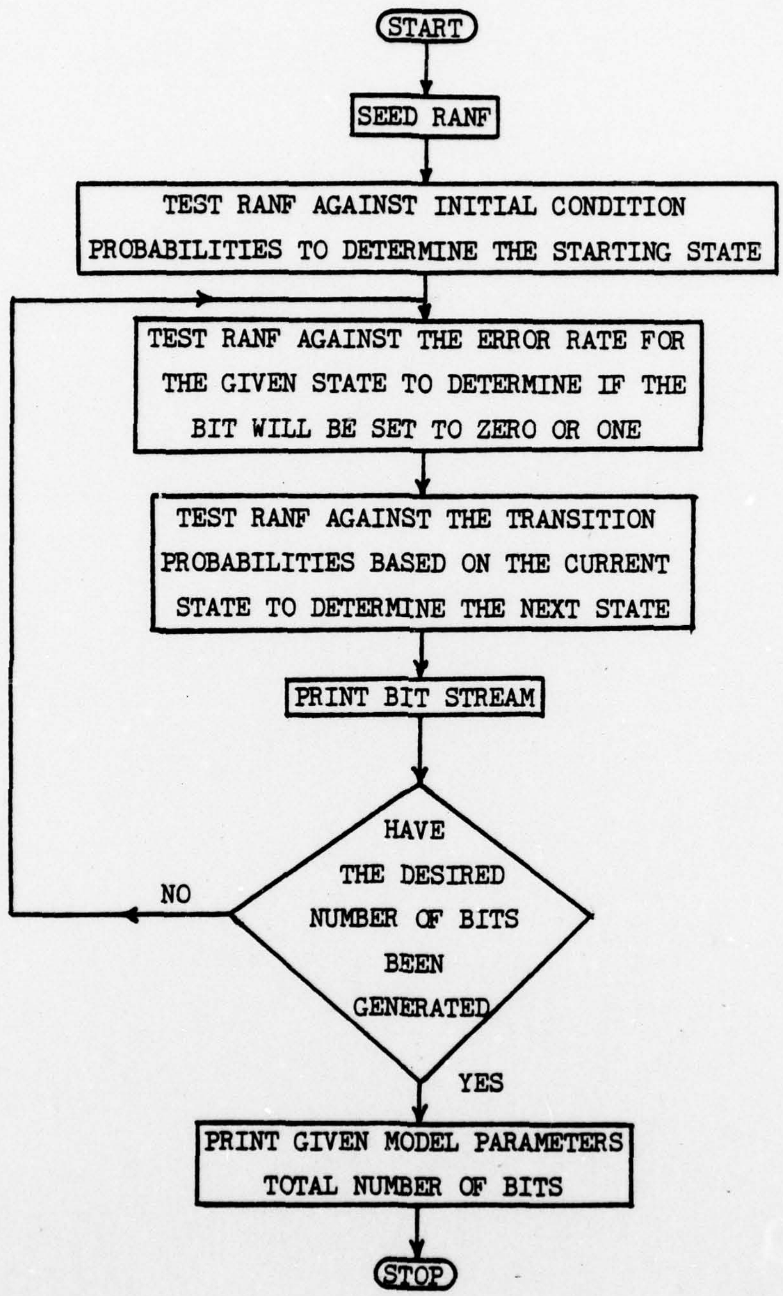
RVE = randomly generated number to determine the bit errors for the given state

C(J) = array of 100 bits generated by the program

M = the number of 100 bit blocks generated

TOTAL = the total number of bits generated

Flow Chart for Burst and Gap Pattern Test Program



```
      PROGRAM THESIS (INPUT=/80,OUTPUT=/132)
      INTEGER C(100)
      READ*,GA,GB,PE1,PE2
      CALL TIME (C)
      CALL RANSET (C)
      RVN=RANF(DUMMY)
      IF(RVN.LE.GA/(GA+GB)) NP=2
      IF(RVN.GT.GA/(GA+GB)) NP=1
      M= 20
      DO 10 L=1,M
      DO 20 J=1,100
      IF(NP.EQ.2) GO TO 40
      RVN=RANF(DUMMY)
      IF(RVN.LE.GA) NP=2
      RVE=RANF(DUMMY)
      IF(RVE.LE.PE1) GO TO 30
      C(J) = 0
      GO TO 20
   30 C(J)= 1
      GO TO 20
   40 RVN=RANF(DUMMY)
      IF(RVN.LE.GB) NP=1
      RVE=RANF(DUMMY)
      IF(RVE.LE.PE2) GO TO 45
      C(J)= 0
      GO TO 20
   45 C(J)= 1
   20 CONTINUE
      WRITE 100, C
  100 FORMAT(1X,50I1/1X,50I1)
   10 CONTINUE
      TOTAL= M*100
      PRINT*," "
      PRINT*," "
      PRINT*,"GIVEN MODEL PARAMETERS"
      PRINT*," "
      PRINT*,"GA= ",GA
      PRINT*,"GB= ",GB
      PRINT*,"PE1= ",PE1
      PRINT*,"PE2= ",PE2
      PRINT*," "
      PRINT*,"NUMBER OF BITS IN THE TEST= ",TOTAL
      PRINT*," "
      END
```

```
00000000000000000010001000000111110000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000101011110011000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
01100011111100000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000001000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000101001100000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

000000000000000101000000000000000000000000000000000
00000000000000000000000000000000001010010001010110

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000110

00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000100
00000000000000000000000000000000000000001001101000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
```

GIVEN MODEL PARAMETERS

GA= .01
GB= .1

PE1= .001
PE2= .5

NUMBER OF BITS IN THE TEST= 2000.

Figure 9. Bit Stream Produced by Base Test Data

```
00000000000000000000100000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000001011000000000001100000000000000000000000
00000000000000000000000000000000000000000000000000
00000001000000000000000000000000000000000000000000
00111110101001000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000110100
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
01110001010011011110000000000000000100000000000000
00000100000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000001
00000000000000000000000000000000000000000000000000
00000000000010010111100101101001010101101000111001
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000001000000000000000000000000
00000000000000000000010010010000000000000000010000
00111100001010111000101100111001110110010011011100
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000010100000000000000000000100000000011
00110011000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000010000000000000000
00000000000000000000000000000000000001000000000000
00110110100011101011101011000000000000011011000000
00000000000000000000000000000000000000000000000100
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00001000000000000000111011111000100000000000000000
00000000000000000000000000000000000000000000000000
```

GIVEN MODEL PARAMETERS

GA= .01
GB= .1
PE1= .01
PE2= .5

NUMBER OF BITS IN THE TEST= 2000.

Figure 10. Bit Stream Produced with Parameter PE1 Changed

```
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000010000000000000000
00101000100000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000100000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00100011100010101100000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000010001001100000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000110101
11010111011000100000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000

GIVEN MODEL PARAMETERS

GA= .001
GB= .1
PE1= .001
PE2= .5

NUMBER OF BITS IN THE TEST= 2000.
```

Figure 11. Bit Stream Produced with Parameter A Changed

```
00010111000011010101100010000100010000010100000001
11010101000000011011101101010100101010101101101000
00011100101011001110101000001011011011101000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000001
00100111010100000000000000000000000000000000000000
10001001011111110101010101001001001000100110101011
11011100100010110100101100010110100010111010011111
01010010111010011100110010000010101001000000000000
00000000000000000000000000000000000000000000000000
00000000010001011101111100010000001010000001011110
00001100000001011111101100010100010111100101001110
00101011001001000011010001001101000011011001111101
10001110100100011110001001101101000100000110000101
10100001001011010100000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000110010100010101101110000000100000110111
00100100010001011100000010100101001111100110001000
11010110111011111100110110010000000000000000000000
00000000000000000000000000111001010100000000011011
11110111011011110101011010001011110111111101011001
11101110100001010101000101001110010101000110110111
00000001100100000100101111010000010011001100000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000001110110101000110100000000000000000000000
00000000010100111100100101011010000101000001 0010
11110110101111100100011110101000000000101100000010
10100000000000000000000000000000000011100001100010 01001
01110100101010011101001101100010001010011010001010
00010001001000010110011000101110010010101110101100
00011001000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000001000100001011110110111100
00011010100110011001000001111111110010001110100001
```

GIVEN MODEL PARAMETERS

GA= .01
GB= .01
PE1= .0 01
PE2= .5

NUMBER OF BITS IN THE TEST= 2000.

Figure 12. Bit Stream Produced with Parameter B Changed

```
00000000000000000000000000000000000000000001100000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000001111111
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000001
11111111110111100000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000101111111111111110000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000101110000011110111111111111111111
11111111111000000000000000000000000000000000000000
00000000000000000000000000000000000111111111011111
11111110111111111111000000000000000000000000000000
00000000000000000000000011111111000000000000000000
00000000000000000000000000000000000000000000000000
00000000010000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000001111000000000111000000000000
00000000000000000000000000000000001000000011000000
00000000000000000000000000000000000000010000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000011111111111111111111110111
11000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000011111111111110000000000000000000
00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000
```

GIVEN MODEL PARAMETERS

GA= .01
GB= .1
PE1= .001
PE2= .9

NUMBER OF BITS IN THE TEST= 2000.

Figure 13. Bit Stream Produced with Parameter PE2 Changed

Appendix B

Program to Test Relative Frequency Estimation Convergence

This program uses the same technique to generate an error bit stream as used in appendix A. It also tests successive bits to count errors, overlapping pairs of errors, and overlapping triple errors. M was varied to test bit streams of 2000 to one-million bits. This program contains symbols defined in appendix A, plus these symbols.

P1 = counter of error bits, then divided by the total number of bits to yield P(1)

PR11 = counter of overlapping pairs of consecutive errors, divided by total - 1 to yield P(1,1)

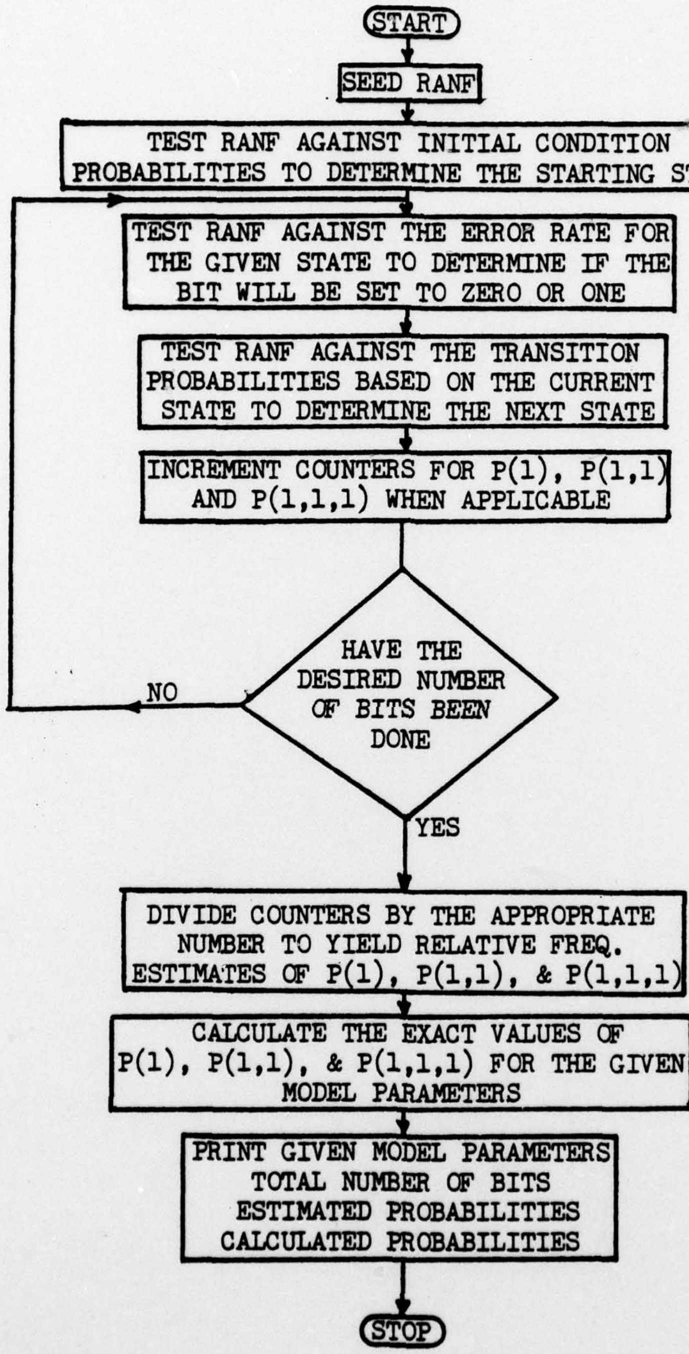
PR111 = counter of overlapping triple consecutive error, divided by total - 2 to yield P(1,1,1)

T1 = temporary storage for the 99^{th} bit in the array so it can be tested with the first bits of the next array

T = temporary storage for the 100^{th} bit in the array so it can be tested with the first bits of the next array

W = used as default so the first time through the loop T & T1 won't be tested against the first array

Flow Chart for Relative Frequency Program



```
      PROGRAM THESIS (INPUT=/80,OUTPUT=/132)
      INTEGER C(100)
      READ*,GA,GB,PE1,PE2
      CALL TIME (C)
      CALL RANSET (C)
      RVN=RANF(DUMMY)
      IF(RVN.LE.GA/(GA+GB)) NP=2
      IF(RVN.GT.GA/(GA+GB)) NP=1
      P1= 0.0
      PR11=0.0
      PR111= 0.0
      T= 0.0
      T1= 0.0
      W=0.0
      M= 20
      DO 10 L=1,M
      DO 20 J=1,100
      IF(NP.EQ.2) GO TO 40
      RVN=RANF(DUMMY)
      IF(RVN.LE.GA) NP=2
      RVE=RANF(DUMMY)
      IF(RVE.LE.PE1) GO TO 30
      C(J) = 0
      GO TO 20
   30 C(J)= 1
      P1= P1 + 1.0
      GO TO 20
   40 RVN=RANF(DUMMY)
      IF(RVN.LE.GB) NP=1
      RVE=RANF(DUMMY)
      IF(RVE.LE.PE2) GO TO 45
      C(J)= 0
      GO TO 20
   45 C(J)= 1
      P1= P1 + 1.0
   20 CONTINUE
      IF(W.EQ.0.0) GO TO 50
      IF(C(1).EQ.0) GO TO 50
      IF(T.EQ.0.0) GO TO 50
      IF(C(1).EQ.1.0) PR11= PR11 + 1.0
      IF(T1.EQ.1) PR111= PR111 + 1.0
      IF(C(2).EQ.1) PR111= PR111 + 1.0
   50 CONTINUE
```

```
      DO 60 K=1,99
      IF(C(K).EQ.0.0) GO TO 60
      IF(C(K+1).EQ.1.0) PR11= PR11 + 1.0
      IF(C(K+1).EQ.0) GO TO 60
      IF(K.EQ.99) GO TO 60
      IF(C(K+2).EQ.1) PR111= PR111 + 1.0
   60 CONTINUE
      IF(M.GT.1) W= 1.0
      T1= C(99)
      T= C(100)
   10 CONTINUE
      TOTAL= M*100
      P1= P1/(TOTAL)
      PR11= PR11/(TOTAL- 1)
      PR111= PR111/(TOTAL-2)
      CALP1= (GB*PE1 + GA*PE2)/(GA+GB)
      CAL11= ((PE1 + GA*(PE2 - PE1))*GB*PE1 +
     %  (PE2 + GB*(PE1 - PE2))*GA*PE2)/
     %  (GA +GB)
      CAL111= ((GB*PE1*(PE1+GA*(PE2-PE1))+(1-GB)*PE2*(
     %  PE2+GB*(PE1-PE2)))*GA*PE2+((1-GA)*PE1*(PE1+GA*(
     %  PE2-PE1))+GA*PE2*(PE2+GB*(PE1-PE2)))*GB*PE1)/(GA+GB)
      PRINT*," "
      PRINT*,"P(1)= ",P1,"    P(1,1)= ",PR11,"    P(1,1,1)= ",PR111
      PRINT*,"CALC P(1)= ",CALP1," CALC P(1,1)= ",
     % CAL11,"   CALC P(1,1,1)= ",CAL111
      PRINT*," "
      PRINT*,"GIVEN MODEL PARAMETERS"
      PRINT*," "
      PRINT*,"GA= ",GA
      PRINT*,"GB= ",GB
      PRINT*,"PE1= ",PE1
      PRINT*,"PE2= ",PE2
      PRINT*," "
      PRINT*,"NUMBER OF BITS IN THE TEST= ",TOTAL
      PRINT*," "
      END
```

Appendix C

Program of Successive Approximation

This program uses a loop of successive approximations of the model parameters based on Eqs (26) and (27) and the independent equations P(1), P(1,1), and P(1,1,1). Symbols are the same as those found in appendix A plus these symbols.

AHAT = equation of P(1) in the form of Eq (26)

BHAT = equation of P(1,1) in the form of Eq (26)

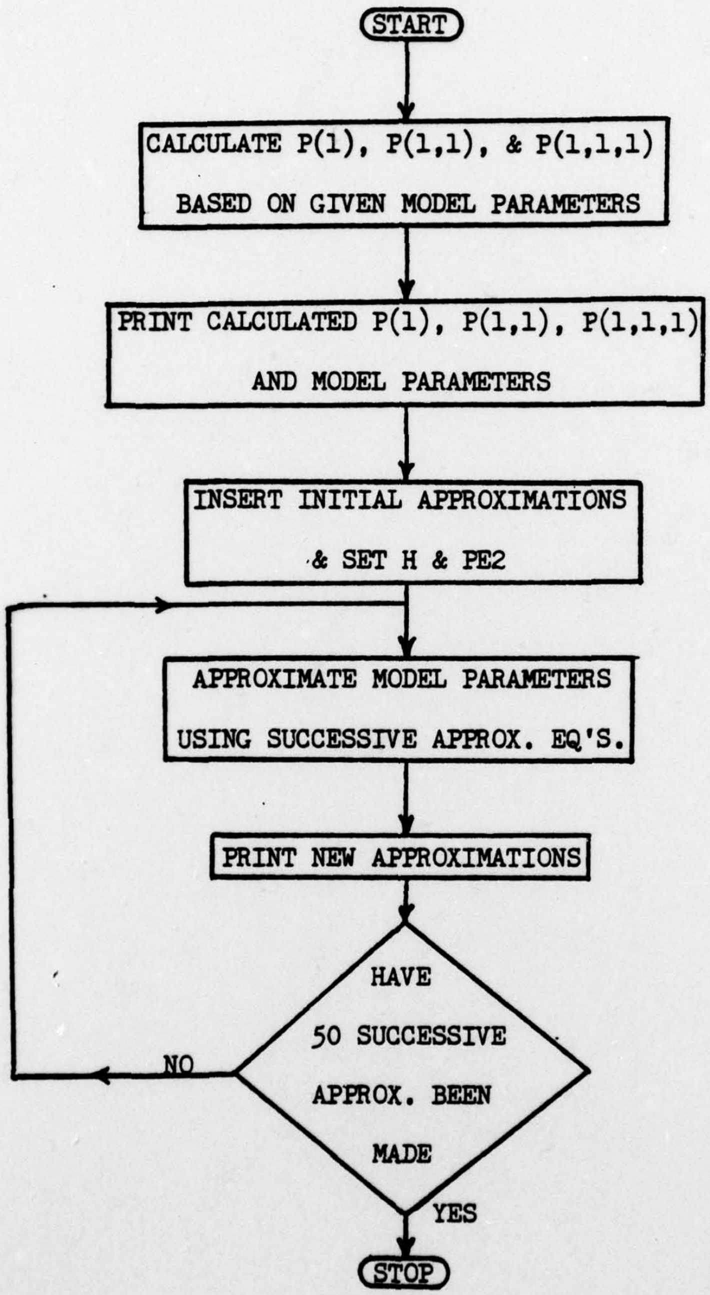
E1HAT = equation of P(1,1,1) in the form of Eq (26)

A = the successive approximation of parameter A using Eq (27)

B = the successive approximation of parameter B using Eq (27)

E1 = the successive approximation of parameter PE1 using Eq (27)

Flow Chart for Successive Approximation Program



```
      PROGRAM THESIS (INPUT=/80,OUTPUT=/132)
      READ*,GA,GB,PE1,PE2
      CALP1= (GB*PE1 + GA*PE2)/(GA+GB)
      CAL11= ((PE1 + GA*(PE2 - PE1))*GB*PE1 +
     % (PE2 + GB*(PE1 - PE2))*GA*PE2)/
     % (GA +GB)
      CAL111= ((GB*PE1*(PE1+GA*(PE2-PE1))+(1-GB)*PE2*(
     % PE2+GB*(PE1-PE2)))*GA*PE2+((1-GA)*PE1*(PE1+GA*(
     % PE2-PE1))+GA*PE2*(PE2+GB*(PE1-PE2)))*GB*PE1)/(GA+GB)
      PRINT*," "
      PRINT*,"CALC P(1)= ",CALP1," CALC P(1,1)= ",
     % CAL11,"    CALC P(1,1,1)= ",CAL111
      PRINT*," "
      PRINT*,"GIVEN MODEL PARAMETERS"
      PRINT*," "
      PRINT*,"GA= ",GA
      PRINT*,"GB= ",GB
      PRINT*,"PE1= ",PE1
      PRINT*,"PE2= ",PE2
      PRINT*," "
      E2= 0.5
      H= 1E-4
      A= 0.009
      B= 0.125
      E1= 0.00125
      DO 110 I= 1,50
      AHAT= (B*E1 + A*E2)/CALP1 - B
      A= (1-H)*A + H*AHAT
      IF(A.GT.B/10.0) A= B/10.0
      IF(A.LT.1E-6) A= 1E-6
      BHAT= ((E1 + A*(E2 - E1))*(B*E1) + (E2 + B*(E1
     % - E2))*(A*E2))/CAL11 - A
      B= (1-H)*B + H*BHAT
      IF(B.GT.0.5) B= 0.5
      IF(B.LT.1E-3) B= 1E-3
      E1HAT=(CAL111*(A+B) - (B*E1*(E1+A*(E2-E1))+(1-B)*
     % E2*(E2+B*(E1-E2)))*A*E2)/((((1-A)*E1*(E1+A*(E2-E1
     % ))+A*E2*(E2+B*(E1-E2)))*B)
      E1=(1-H)*E1 + H*E1HAT
      IF(E1.GT.E2/100.0) E1= E2/100.0
      IF(E1.LT.1E-6) E1= 1E-6
      PRINT*,"AHAT= ",A,"    BHAT= ",B,"    E1HAT= ",E1,"   H= ",H
  110 CONTINUE
      END
```

Vita

Robert P. Davis was born August 5, 1947 in Rochester, New York. He graduated from high school in Churchville, New York in 1965, and then attended Grove City College in Grove City, Pennsylvania from which he received the degree of Bachelor of Science in Mechanical Engineering and a commission in the United States Air Force in 1969. He attended undergraduate pilot training at Williams AFB, Arizona, and received the aeronautical rating of pilot in 1970. He married Nancy Jeffers on September 19, 1970. From September, 1970 to May, 1976, Capt. Davis was assigned as a B-52 pilot in the 19th Bombardment Wing, Robins AFB, Georgia. During that time, he served 429 days TDY in Southeast Asia and flew'198 combat missions. Capt. Davis entered the Air Force Institute of Technology in June 1976. He is a member of Eta Kappa Nu and Tau Beta Pi.

Permanent address 2027 Westside Drive
Rochester, New York 14624

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/78-24 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A BURST ERROR MODEL USING INTERLEAVED BERNOULLI PROCESSES WITH A MARKOV PROPERTY		5. TYPE OF REPORT & PERIOD COVERED M.S. Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert P. Davis		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQ AFCS/EETDD Scott AFB IL 62225		12. REPORT DATE December 1978 13. NUMBER OF PAGES 76
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR190-17
Joseph P. Hipps, Major, USAF
Director of Information
1-23-79

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Burst Error Modeling
Channel Model
Digital Communications
Mathematical Modeling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A model of burst errors from a real-time error bit stream of up to 20 million bits per second is sought. Noise models and channel models are considered. The choice is a two-state Markov channel model in which the states each generate independent Bernoulli random processes. One Bernoulli process produces ones (errors) at a high rate, simulating burst errors. The other process produces ones at a low rate, simulating random errors due to background noise. The transition probabilities determine the average length of the bursts and of the gaps. Relative frequency estimates

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20

of the probabilities of certain sequences of one-bits from real data are related to estimates of the model parameters, so relative frequencies provide a basis for fitting this model to real channels using observed error sequences. An equation for the number of errors in a block of bits is developed in terms of the model parameters. Burst probabilities can be predicted based on this equation. The model was tested using computer simulation. Some discussion is devoted to how this burst-error model can be implemented in an actual device to provide real-time channel characterizations. This model aids in the selection of an error correction code.